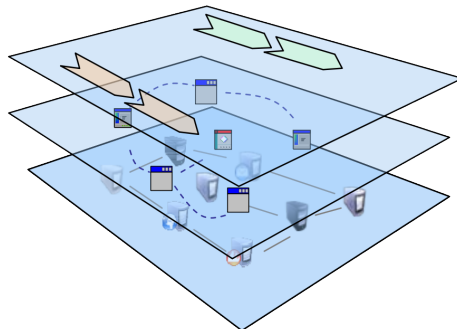


Klassenbeschreibung

3. September 2008
Projektgruppe Modulares EAM System

Christian Rolfes, Christian Zillmann, David Heimann,
Hyung-Bin Kim, Igor Sechyn, Jens Henkel, Jörn Trefke,
Mart Köhler, Philipp Gringel, Roland Koppe, Yu Liu



Carl von Ossietzky Universität Oldenburg
Fakultät II - Department für Informatik
Abteilung Informationssysteme - Prof. Dr. Dr. h.c. H.-J. Appelrath

OFFIS - Institut für Informatik
Bereich Betriebliches Informationsmanagement

Inhaltsverzeichnis

1	de.offis.pg.eam.core	3
1.1	BundleConfiguration	3
1.1.1	BundleViews Klassenreferenz	3
1.2	auth	3
1.2.1	IAuthNames Schnittstellenreferenz	3
1.2.2	IDalBundleManager Schnittstellenreferenz	3
1.2.3	IDalGroup Schnittstellenreferenz	4
1.2.4	IDalRole Schnittstellenreferenz	4
1.2.5	IDalUser Schnittstellenreferenz	4
1.2.6	IMetamodelAuth Schnittstellenreferenz	4
1.2.7	IUserInformation Schnittstellenreferenz	4
1.2.8	AuthNames Klassenreferenz	4
1.2.9	Dal Klassenreferenz	6
1.2.10	DalBundleManager Klassenreferenz	9
1.2.11	DalGroup Klassenreferenz	35
1.2.12	DalRole Klassenreferenz	40
1.2.13	DalUser Klassenreferenz	49
1.2.14	MetamodelAuth Klassenreferenz	55
1.2.15	UserInformation Klassenreferenz	60
1.2.16	AuthInfo Klassenreferenz	60
1.2.17	Group Klassenreferenz	61
1.2.18	LoginTag Klassenreferenz	61
1.2.19	RoleTag Klassenreferenz	62
1.2.20	UMUser Klassenreferenz	63
1.2.21	UserManager Klassenreferenz	63
1.3	beans	72
1.3.1	UserInfoBean Klassenreferenz	72
1.3.2	UserManagerBean Klassenreferenz	72
1.3.3	BundleInformationBean Klassenreferenz	72
1.3.4	ContainerBean Klassenreferenz	72
1.3.5	ImportExportBean Klassenreferenz	73
1.3.6	LoggingBean Klassenreferenz	73
1.3.7	ResourceServlet Klassenreferenz	74
1.3.8	Resources Klassenreferenz	74
1.3.9	MultipartWrapper Klassenreferenz	76
1.3.10	Part Klassenreferenz	77
1.3.11	UploadServlet Klassenreferenz	80
1.4	configuration	81

1.4.1	IBundleConfiguration Schnittstellenreferenz	81
1.4.2	JdomXMLFileReader Klassenreferenz	81
1.4.3	EAMObjectAttributeMatchingStore Klassenreferenz	82
1.4.4	EAMObjectMatchingStore Klassenreferenz	83
1.4.5	EAMRelationAttributeMatchingStore Klassenreferenz	83
1.4.6	EAMRelationMatchingStore Klassenreferenz	83
1.4.7	EAMViewRoleMatchingStore Klassenreferenz	84
1.4.8	ModuleToCoreObjectWrapper Klassenreferenz	84
1.5	container	88
1.5.1	ContainerHandlerMemory Klassenreferenz	88
1.5.2	ContainerHandlerSql Klassenreferenz	89
1.5.3	ObjectWrapper Klassenreferenz	90
1.5.4	ContainerObject Klassenreferenz	90
1.6	database	91
1.6.1	DatabaseConnectionException Klassenreferenz	91
1.6.2	DatabaseException Klassenreferenz	91
1.6.3	DatabaseResultSetException Klassenreferenz	91
1.6.4	DatabaseSQLException Klassenreferenz	92
1.6.5	DatabaseStatementException Klassenreferenz	92
1.6.6	Connection Klassenreferenz	92
1.6.7	ConnectionHolder Klassenreferenz	96
1.6.8	Database Klassenreferenz	97
1.6.9	Column Klassenreferenz	100
1.7	hibernate	100
1.7.1	HibernateJPAUtil Klassenreferenz	100
1.7.2	HibernateTransactionServletfilter Klassenreferenz	101
1.7.3	Setup Klassenreferenz	102
1.8	import_export	102
1.8.1	ErrorHandler Klassenreferenz	102
1.8.2	ExportAttributeDataTypeMissingException Klassenreferenz	102
1.8.3	ExportAttributeTypeMissingException Klassenreferenz	102
1.8.4	ExportException Klassenreferenz	103
1.8.5	ExportFirstRelationMemberMissingException Klassenreferenz	103
1.8.6	ExportSecondRelationMemberMissingException Klassenreferenz	103
1.8.7	ImportException Klassenreferenz	104
1.8.8	ImportInstancesException Klassenreferenz	104
1.8.9	ImportIOException Klassenreferenz	104
1.8.10	ImportMetamodelException Klassenreferenz	104
1.8.11	ImportParseException Klassenreferenz	105
1.8.12	Export Klassenreferenz	105
1.8.13	ExportInstances Klassenreferenz	107
1.8.14	ExportMetamodel Klassenreferenz	107
1.8.15	AnalyseInstances Klassenreferenz	108
1.8.16	AnalyseInstancesContent Klassenreferenz	108
1.8.17	AnalyseMetamodel Klassenreferenz	109
1.8.18	AnalyseMetamodelContent Klassenreferenz	109

1.8.19	Import Klassenreferenz	109
1.8.20	ObjectInstance Klassenreferenz	109
1.8.21	ObjectInstances Klassenreferenz	110
1.8.22	RelationInstance Klassenreferenz	110
1.8.23	RelationInstances Klassenreferenz	110
1.8.24	IEDefinition Klassenreferenz	110
1.8.25	IEInstancesDocument Klassenreferenz	110
1.8.26	IEMetamodelDocument Klassenreferenz	111
1.8.27	ExportModel Klassenreferenz	111
1.8.28	IEAttribute Klassenreferenz	112
1.8.29	IEAttributeType Klassenreferenz	112
1.8.30	IEObject Klassenreferenz	112
1.8.31	IEObjectType Klassenreferenz	112
1.8.32	IERelation Klassenreferenz	113
1.8.33	IERelationType Klassenreferenz	113
1.8.34	IEStatus Klassenreferenz	113
1.8.35	IESuperClass Klassenreferenz	113
1.8.36	IETime Klassenreferenz	114
1.8.37	IEType Klassenreferenz	114
1.8.38	ImportModel Klassenreferenz	114
1.8.39	Model Klassenreferenz	114
1.9	logging	117
1.9.1	ILoggingDAO Schnittstellenreferenz	117
1.9.2	Logger Klassenreferenz	117
1.9.3	LoggingDAO Klassenreferenz	120
1.9.4	Log Klassenreferenz	120
1.10	menu	120
1.10.1	AbstractAdoptableMenuContributor Klassenreferenz	120
1.10.2	AbstractMenuContributor Klassenreferenz	123
1.10.3	AbstractMenuContributorAdopter Klassenreferenz	125
1.10.4	AuthMenuItem Klassenreferenz	127
1.10.5	IMenuContributor Schnittstellenreferenz	130
1.10.6	IMenuContributorAdopter Schnittstellenreferenz	132
1.10.7	IMenuItem Schnittstellenreferenz	134
1.10.8	ITopMenuConstants Schnittstellenreferenz	139
1.10.9	MenuContributorUtils Klassenreferenz	141
1.10.10	MenuItem Klassenreferenz	141
1.10.11	PageAccessTrackServlet Klassenreferenz	146
1.10.12	SimpleMenuItem Klassenreferenz	150
1.10.13	Submenu Klassenreferenz	152
1.10.14	SysMsgManager Klassenreferenz	155
1.10.15	CoreServer Klassenreferenz	156
1.10.16	DefaultMenuVisitor Klassenreferenz	160
1.10.17	FacesServletWrapper Klassenreferenz	161
1.10.18	IAccessRight Schnittstellenreferenz	163
1.10.19	IMenuVisitor Schnittstellenreferenz	164

1.10.20	ITopMenuUpdateListener	Schnittstellenreferenz	165
1.10.21	JddmMenuParamsGenerator	Klassenreferenz	165
1.10.22	ModuleEvent	Klassenreferenz	167
1.10.23	ModuleEventDispatcher	Klassenreferenz	171
1.10.24	ModuleEventHandler	Schnittstellenreferenz	172
1.10.25	ModuleInstalledEvent	Klassenreferenz	173
1.10.26	ModuleStartedEvent	Klassenreferenz	174
1.10.27	ModuleStoppedEvent	Klassenreferenz	174
1.10.28	ModuleUninstalledEvent	Klassenreferenz	174
1.10.29	Notification	Klassenreferenz	175
1.10.30	PrintfFormat	Klassenreferenz	175
1.10.31	SessionClosed	Klassenreferenz	183
1.10.32	SessionManager	Klassenreferenz	184
1.10.33	SysEvent	Klassenreferenz	185
1.10.34	SysMessage	Klassenreferenz	189
1.10.35	SysMsgServerThread	Klassenreferenz	191
1.10.36	TopMenu	Klassenreferenz	191
1.10.37	TopMenuBar	Klassenreferenz	195
1.10.38	TopMenuServerThread	Klassenreferenz	196
1.11	metamodel		197
1.11.1	CheckRights	Klassenreferenz	197
1.11.2	Validator	Klassenreferenz	198
1.11.3	InstanceAttributeDataTypeDomainException	Klassenreferenz	202
1.11.4	InstanceAttributeDataTypeMissingException	Klassenreferenz	202
1.11.5	InstanceAttributeDomainException	Klassenreferenz	202
1.11.6	InstanceAttributeEnumDomainException	Klassenreferenz	203
1.11.7	InstanceAttributeException	Klassenreferenz	203
1.11.8	InstanceAttributeNotUniqueValueException	Klassenreferenz	203
1.11.9	InstanceAttributeRegexException	Klassenreferenz	204
1.11.10	InstanceAttributeTypeDomainException	Klassenreferenz	204
1.11.11	InstanceAttributeTypeMissingException	Klassenreferenz	204
1.11.12	InstanceAttributeValueIncompatibleException	Klassenreferenz	204
1.11.13	InstanceAttributeValueNotNumericException	Klassenreferenz	205
1.11.14	InstanceDatabaseException	Klassenreferenz	205
1.11.15	InstanceDataObjectDeleteException	Klassenreferenz	205
1.11.16	InstanceDataObjectSaveException	Klassenreferenz	205
1.11.17	InstanceException	Klassenreferenz	206
1.11.18	InstanceIsNullException	Klassenreferenz	206
1.11.19	InstanceLockingException	Klassenreferenz	206
1.11.20	InstanceLockingInUseException	Klassenreferenz	206
1.11.21	InstanceMetaObjectException	Klassenreferenz	207
1.11.22	InstanceMetaObjectHasNoAttributesException	Klassenreferenz	207
1.11.23	InstanceMetaObjectInvalidException	Klassenreferenz	207
1.11.24	InstanceMetaObjectIsNullException	Klassenreferenz	208
1.11.25	InstanceRegexException	Klassenreferenz	208
1.11.26	InstanceRelationException	Klassenreferenz	208

1.11.27	InstanceRelationFirstInvalidException	Klassenreferenz	208
1.11.28	InstanceRelationFirstMissingException	Klassenreferenz	209
1.11.29	InstanceRelationFirstMultiplicityException	Klassenreferenz	209
1.11.30	InstanceRelationMultiplicityException	Klassenreferenz	209
1.11.31	InstanceRelationSecondInvalidException	Klassenreferenz	209
1.11.32	InstanceRelationSecondMissingException	Klassenreferenz	210
1.11.33	InstanceRelationSecondMultiplicityException	Klassenreferenz	210
1.11.34	InstanceRightsMissingException	Klassenreferenz	210
1.11.35	InstanceSQLInvalidException	Klassenreferenz	210
1.11.36	InstanceSQLTableNotExistsException	Klassenreferenz	211
1.11.37	InstanceTableAttributeDataTypeDomainException	Klassenreferenz	211
1.11.38	InstanceTableAttributeDataTypeMissingException	Klassenreferenz	211
1.11.39	InstanceTableAttributeDefaultValueIncompatibleException	Klassenreferenz	211
1.11.40	InstanceTableAttributeIncompatibleException	Klassenreferenz	212
1.11.41	InstanceTableAttributeTypeMissingException	Klassenreferenz	212
1.11.42	InstanceTableDeleteException	Klassenreferenz	212
1.11.43	InstanceTableEnumRegexException	Klassenreferenz	212
1.11.44	InstanceTableEnumRegexMissingException	Klassenreferenz	213
1.11.45	InstanceTableException	Klassenreferenz	213
1.11.46	InstanceTableSaveException	Klassenreferenz	213
1.11.47	DataObjectList	Klassenreferenz	214
1.11.48	IInstanceDataInterface	Schnittstellenreferenz	219
1.11.49	IInstanceQuery	Schnittstellenreferenz	219
1.11.50	InstanceDataInterface	Klassenreferenz	219
1.11.51	InstanceInterface	Klassenreferenz	221
1.11.52	InstanceLoader	Klassenreferenz	222
1.11.53	InstanceLoaderSQL	Klassenreferenz	223
1.11.54	InstanceMetaInterface	Klassenreferenz	224
1.11.55	InstanceQuery	Klassenreferenz	225
1.11.56	Query	Klassenreferenz	228
1.11.57	Lock	Klassenreferenz	231
1.11.58	Locking	Klassenreferenz	233
1.11.59	DataObject	Klassenreferenz	235
1.11.60	DataObjectBean	Klassenreferenz	239
1.11.61	DataObjectScrollerTag	Klassenreferenz	239
1.11.62	DataObjectTableTag	Klassenreferenz	242
1.11.63	DAOFactory	Klassenreferenz	244
1.11.64	HibernateDAOFactoryImpl	Klassenreferenz	247
1.11.65	EAMObjectDAOHibernate	Klassenreferenz	248
1.11.66	EAMRelationDAOHibernate	Klassenreferenz	248
1.11.67	GenericHibernateDAOImpl< T, ID extends Serializable >	Klassenreferenz	250
1.11.68	AuthDAO	Schnittstellenreferenz	252
1.11.69	EAMObjectDAO	Schnittstellenreferenz	253
1.11.70	GenericDAO< T, ID extends Serializable >	Schnittstellenreferenz	254

1.11.71	DAOFactory	Klassenreferenz	256
1.11.72	HibernateDAOFactoryImpl	Klassenreferenz	259
1.11.73	GenericDAO< T, ID extends Serializable >	Schnittstellenreferenz	259
1.11.74	MetaInterface	Klassenreferenz	261
1.11.75	MetamodelHelper	Klassenreferenz	263
1.11.76	IQueryDAO	Schnittstellenreferenz	264
1.11.77	QueryDAO	Klassenreferenz	264
1.11.78	Query	Klassenreferenz	268
1.12	module		268
1.12.1	IModuleView	Schnittstellenreferenz	268
1.13	property		269
1.13.1	EAMProperties	Klassenreferenz	269
1.14	proxy		271
1.14.1	EAMProxy	Klassenreferenz	271
1.14.2	Crawler	Klassenreferenz	272
1.14.3	CrawlerCore	Klassenreferenz	274
1.14.4	MethodCrawler	Klassenreferenz	274
1.14.5	MethodDescription	Klassenreferenz	274
1.14.6	MethodDescriptionCrawler	Klassenreferenz	274
1.14.7	Crawler	Klassenreferenz	275
1.14.8	CrawlerCore	Klassenreferenz	276
1.14.9	MethodCrawler	Klassenreferenz	277
1.14.10	MethodDescription	Klassenreferenz	277
1.14.11	MethodDescriptionCrawler	Klassenreferenz	277
1.14.12	ProxyException	Klassenreferenz	278
1.14.13	ProxyImplementationClassException	Klassenreferenz	278
1.14.14	ProxyInterfaceException	Klassenreferenz	278
1.14.15	ProxyInvocationException	Klassenreferenz	279
1.14.16	ProxyInvocationUserHasNoRightsException	Klassenreferenz	279
1.14.17	ProxyInvocationUserNotExistsException	Klassenreferenz	279
1.14.18	ProxyInterface	Schnittstellenreferenz	279
1.14.19	ProxyInterfaceBuilder	Klassenreferenz	280
1.14.20	IAuthUser	Schnittstellenreferenz	280
1.15	test		281
1.15.1	LockThread	Klassenreferenz	281
1.15.2	Test_AuthNames	Klassenreferenz	281
1.15.3	Test_Database	Klassenreferenz	281
1.15.4	Test_DatabaseInsert	Klassenreferenz	281
1.15.5	Test_DataObject	Klassenreferenz	282
1.15.6	Test_DataObjectList	Klassenreferenz	282
1.15.7	Test_DataTypeValidator	Klassenreferenz	282
1.15.8	Test_DOInsertVerantwortlicher	Klassenreferenz	282
1.15.9	Test_EAMProperties	Klassenreferenz	282
1.15.10	Test_EAMProxy	Klassenreferenz	283
1.15.11	Test_Hibernate	Klassenreferenz	283
1.15.12	Test_ImportExport	Klassenreferenz	283

1.15.13	Test_Instance_Refactor1	Klassenreferenz	283
1.15.14	Test_InstanceDataInterface	Klassenreferenz	284
1.15.15	Test_InstanceDelete	Klassenreferenz	284
1.15.16	Test_InstanceInterface	Klassenreferenz	284
1.15.17	Test_InstanceLocking	Klassenreferenz	284
1.15.18	Test_InstanceLockingThread	Klassenreferenz	285
1.15.19	Test_InstanceQuery	Klassenreferenz	285
1.15.20	Test_LoggingFiles	Klassenreferenz	285
1.15.21	Test_MetamodelAuth	Klassenreferenz	285
1.15.22	Test_MetamodelHelper	Klassenreferenz	285
1.15.23	Test_SQL_Split	Klassenreferenz	286
1.15.24	Test_SQL_Escape	Klassenreferenz	286
1.15.25	Test_XMLValidator	Klassenreferenz	286
1.16	tools		286
1.16.1	File	Klassenreferenz	286
1.16.2	SQLTools	Klassenreferenz	288
1.16.3	MessageTag	Klassenreferenz	288
1.16.4	Text	Klassenreferenz	289
1.16.5	ErrorHandler	Klassenreferenz	290
1.16.6	Validator	Klassenreferenz	290
1.16.7	XMLDefinition	Klassenreferenz	291
1.16.8	XMLTools	Klassenreferenz	292
1.16.9	Zip	Klassenreferenz	293
2	de.offis.pg.eam.bundlemanager		295
2.1	BundleConfiguration		295
2.1.1	BundleViews	Klassenreferenz	295
2.1.2	Configuration	Klassenreferenz	296
2.1.3	AbstractBundleManager	Klassenreferenz	297
2.1.4	Activator	Klassenreferenz	301
2.1.5	Activator.HttpServiceTracker	Klassenreferenz	303
2.1.6	Activator.ServletContextListenerServletAdaptor	Klassenreferenz	303
2.1.7	BundleManager	Klassenreferenz	306
2.1.8	Context	Klassenreferenz	318
2.2	Dependency		319
2.2.1	Bundle	Klassenreferenz	319
2.2.2	BundleConnection	Klassenreferenz	322
2.2.3	BundleNode	Klassenreferenz	324
2.2.4	BundleNode.BundleNodeMouseMotion	Klassenreferenz	330
2.2.5	BundleNode.BundleNodeMouseProcess	Klassenreferenz	330
2.2.6	DependencyApplet	Klassenreferenz	331
2.2.7	DependencyPanel	Klassenreferenz	334
2.2.8	DependencyPanel.TimerHandler	Klassenreferenz	339
2.2.9	GraphicsUtils	Klassenreferenz	340
2.2.10	BundleDependencyUtils	Klassenreferenz	342
2.2.11	BundleUsers	Klassenreferenz	346

2.2.12	ImageShow Klassenreferenz	352
2.2.13	ParamsGenerator Klassenreferenz	353
2.3	Matchinghandler	356
2.3.1	BundleConfigurationNavigation Klassenreferenz	356
2.3.2	CoreBundleObjectAttributeMatchingHandler Klassenreferenz	363
2.3.3	CoreBundleObjectMatchingHandler Klassenreferenz	380
2.3.4	CoreBundleRelationAttributeMatchingHandler Klassenreferenz	386
2.3.5	CoreBundleRelationMatchingHandler Klassenreferenz	402
2.3.6	CoreBundleViewRoleMatchingHandler Klassenreferenz	421
2.3.7	DAOFactoryStaticClass Klassenreferenz	437
2.4	Sonstiges	440
2.4.1	EAMBundle Klassenreferenz	440
2.4.2	BundleMgrContributor Klassenreferenz	445
2.4.3	Upload Klassenreferenz	447
3	de.offis.pg.eam.core_usermgr	449
3.1	BundleConfiguration	449
3.1.1	BundleViews Klassenreferenz	449
3.1.2	Configuration Klassenreferenz	450
3.1.3	Activator Klassenreferenz	451
3.1.4	Activator.HttpServiceTracker Klassenreferenz	452
3.1.5	Activator.ServletContextListenerServletAdaptor Klassenreferenz	453
3.1.6	CoreUserMgrContributor Klassenreferenz	456
3.2	Managed Beans	457
3.2.1	GroupOverview Klassenreferenz	457
3.2.2	ManageMenu Klassenreferenz	469
3.2.3	RoleOverview Klassenreferenz	478
3.2.4	UserOverview Klassenreferenz	538
3.3	Sonstiges	551
3.3.1	ExceptionHandlerListener Klassenreferenz	551
3.3.2	File Klassenreferenz	551
4	de.offis.pg.eam.core_datamgr	553
4.1	BundleConfiguration	553
4.1.1	BundleViews Klassenreferenz	553
4.1.2	Configuration Klassenreferenz	554
4.1.3	Activator Klassenreferenz	555
4.1.4	Activator.HttpServiceTracker Klassenreferenz	556
4.1.5	Activator.ServletContextListenerServletAdaptor Klassenreferenz	557
4.1.6	DataMgrContributor Klassenreferenz	560
4.2	Instanzeneingabe	561
4.2.1	DataObjectManagedBean Klassenreferenz	561
4.2.2	QueryResultBean Klassenreferenz	576
4.2.3	RelationDataObjectBean Klassenreferenz	579
4.3	Tag-Library	581
4.3.1	DataObjectScrollerTag Klassenreferenz	581

4.3.2	dataObjectTable Klassenreferenz	587
4.3.3	insertDataObject Klassenreferenz	589
4.3.4	MessageTag Klassenreferenz	591
4.4	Metamodelleingabe	592
4.4.1	EAMAttributeManagedBean Klassenreferenz	592
4.4.2	EAMCategoryManagedBean Klassenreferenz	612
4.4.3	EAMObjectRelationManagedBean Klassenreferenz	620
4.4.4	EAMStaticClass Klassenreferenz	669
5	de.offis.pg.eam.mod_extendDataInput	675
5.1	BundleConfiguration	675
5.1.1	BundleViews Klassenreferenz	675
5.1.2	Configuration Klassenreferenz	676
5.1.3	Activator Klassenreferenz	677
5.1.4	Activator.HttpServiceTracker Klassenreferenz	679
5.1.5	Activator.ServletContextListenerServletAdaptor Klassenreferenz . . .	679
5.1.6	MenuContributor Klassenreferenz	682
5.1.7	MenuConstants Schnittstellenreferenz	683
5.2	Beans	683
5.2.1	AttributeBean Klassenreferenz	683
5.2.2	InstanceBean Klassenreferenz	685
5.2.3	RelationBean Klassenreferenz	689
5.3	Managed Beans	696
5.3.1	ObjectInstanceBean Klassenreferenz	696
5.3.2	ObjectInstancesBean Klassenreferenz	703
5.3.3	RelationInstanceBean Klassenreferenz	715
5.3.4	RelationInstancesBean Klassenreferenz	722
5.4	Tag-Library	735
5.4.1	AttributesTag Klassenreferenz	735
5.4.2	EditAttributeTag Klassenreferenz	737
5.4.3	MessageTag Klassenreferenz	741
5.5	Sonstiges	742
5.5.1	Tools Klassenreferenz	742
6	de.offis.pg.eam.mod_querybrowser	743
6.0.2	BundleViews Klassenreferenz	743
6.0.3	Configuration Klassenreferenz	744
6.0.4	Activator Klassenreferenz	745
6.0.5	Activator.HttpServiceTracker Klassenreferenz	747
6.0.6	Activator.ServletContextListenerServletAdaptor Klassenreferenz . . .	747
6.0.7	MenuConstants Schnittstellenreferenz	750
6.0.8	MenuContributor Klassenreferenz	750
6.1	Managed Beans	751
6.1.1	Query Klassenreferenz	751
6.1.2	QueryBrowserBean Klassenreferenz	752
6.1.3	QueryEditorBean Klassenreferenz	755

6.1.4	QueryResultBean Klassenreferenz	764
6.2	Tag-Library	765
6.2.1	DataObjectScrollerTag Klassenreferenz	765
6.2.2	DataObjectTable Klassenreferenz	765
6.2.3	DataObjectTableTag Klassenreferenz	769
7	de.offis.pg.eam.mod_exp_jasper	773
7.1	BundleConfiguration	773
7.1.1	BundleViews Klassenreferenz	773
7.1.2	Configuration Klassenreferenz	774
7.1.3	BundleProperties Klassenreferenz	775
7.1.4	Activator Klassenreferenz	776
7.1.5	Activator.HttpServiceTracker Klassenreferenz	778
7.1.6	Activator.ServletContextListenerServletAdaptor Klassenreferenz	779
7.1.7	TopMenuBar Klassenreferenz	781
7.1.8	Menu Klassenreferenz	782
7.2	Datenbank	783
7.2.1	Connection Klassenreferenz	783
7.2.2	ConnectionHolder Klassenreferenz	789
7.2.3	Database Klassenreferenz	790
7.2.4	Design Klassenreferenz	795
7.3	Texteditor	797
7.3.1	TextEditor Klassenreferenz	797
7.3.2	TextEditorBean Klassenreferenz	801
7.4	Export-Implementierung	804
7.4.1	ContainerAdapter Klassenreferenz	804
7.4.2	IContainerAdapter Schnittstellenreferenz	807
7.4.3	AbstractOutputFactory Klassenreferenz	807
7.4.4	Context Klassenreferenz	811
7.4.5	CsvExporter Klassenreferenz	811
7.4.6	HtmlExporter Klassenreferenz	812
7.4.7	RtfExporter Klassenreferenz	812
7.4.8	TxtExporter Klassenreferenz	813
7.4.9	XlsExporter Klassenreferenz	813
7.4.10	XmlExporter Klassenreferenz	814
7.4.11	IExporter Schnittstellenreferenz	814
7.4.12	Init Klassenreferenz	815
7.4.13	OutputFactory Klassenreferenz	816
7.4.14	Report Klassenreferenz	818
7.4.15	ReportFrontend Klassenreferenz	826
7.4.16	ReportTest Klassenreferenz	830
7.4.17	MessageTag Klassenreferenz	830
8	de.offis.pg.eam.mod_analysis	833
8.1	BundleConfiguration	833
8.1.1	BundleViews Klassenreferenz	833

8.1.2	Configuration Klassenreferenz	834
8.1.3	Activator Klassenreferenz	835
8.2	API	836
8.2.1	AnalysisFactory Klassenreferenz	836
8.2.2	IDataCollector Schnittstellenreferenz	837
8.2.3	IXmlConverter Schnittstellenreferenz	841
8.2.4	Test Klassenreferenz	842
8.2.5	TreeObject Klassenreferenz	842
8.3	Sonstiges	845
8.3.1	CharacteristicAnalysis Klassenreferenz	845
8.3.2	DataCollector Klassenreferenz	848
8.3.3	XmlConverter Klassenreferenz	858
9	de.offis.pg.eam.mod_vis_1	863
9.1	BundleConfiguration	863
9.1.1	BundleViews Klassenreferenz	863
9.1.2	Configuration Klassenreferenz	864
9.1.3	Activator Klassenreferenz	865
9.1.4	Activator.HttpServiceTracker Klassenreferenz	866
9.1.5	Activator.ServletContextListenerServletAdaptor Klassenreferenz . . .	867
9.1.6	ModVis1MenuContributor Klassenreferenz	870
9.2	Schnittstelle zur Analyse	871
9.2.1	AnalysisAdapter Klassenreferenz	871
9.2.2	IAnalysisAdapter Schnittstellenreferenz	872
9.3	Hilfsklassen	873
9.3.1	Constants Klassenreferenz	873
9.3.2	Definitions Klassenreferenz	875
9.3.3	ExportSVG Klassenreferenz	879
9.3.4	StaxUtil Klassenreferenz	880
9.3.5	Util Klassenreferenz	886
9.4	Implementierung	888
9.4.1	Generator Klassenreferenz	888
9.4.2	Relation Klassenreferenz	892
9.4.3	MatrixVisualization Klassenreferenz	893
9.4.4	SelectObject Klassenreferenz	898
9.4.5	TreeVisualization Klassenreferenz	908
10	de.offis.pg.eam.mod_vis_2	915
10.1	BundleConfiguration	915
10.1.1	BundleViews Klassenreferenz	915
10.1.2	Configuration Klassenreferenz	916
10.1.3	Activator Klassenreferenz	917
10.1.4	Activator.HttpServiceTracker Klassenreferenz	918
10.1.5	Activator.ServletContextListenerServletAdaptor Klassenreferenz . . .	919
10.1.6	ModVis2MenuContributor Klassenreferenz	922
10.2	Schnittstelle zur Analyse	923

10.2.1	AnalysisAdapter Klassenreferenz	923
10.2.2	IAnalysisAdapter Schnittstellenreferenz	924
10.3	Hilfsklassen	925
10.3.1	Constants Klassenreferenz	925
10.3.2	Util Klassenreferenz	926
10.4	Implementierung	927
10.4.1	Generator Klassenreferenz	927
10.4.2	Relation Klassenreferenz	929
10.4.3	SelectObject Klassenreferenz	930
10.4.4	TableVisualization Klassenreferenz	940
11	de.offis.pg.eam.mod_vis_3	945
11.1	BundleConfiguration	945
11.1.1	BundleViews Klassenreferenz	945
11.1.2	Configuration Klassenreferenz	946
11.1.3	Activator Klassenreferenz	947
11.1.4	Activator.HttpServiceTracker Klassenreferenz	948
11.1.5	Activator.ServletContextListenerServletAdaptor Klassenreferenz . . .	949
11.1.6	ModVis3MenuContributor Klassenreferenz	952
11.2	Schnittstelle zur Analyse	953
11.2.1	AnalysisAdapter Klassenreferenz	953
11.2.2	IAnalysisAdapter Schnittstellenreferenz	954
11.3	Hilfsklassen	955
11.3.1	Constants Klassenreferenz	955
11.3.2	ExportSVG Klassenreferenz	956
11.3.3	Util Klassenreferenz	959
11.4	Implementierung	961
11.4.1	MyGraph Klassenreferenz	961
11.4.2	SelectObject Klassenreferenz	965
12	de.offis.pg.eam.mod_example	975
12.1	BundleConfiguration	975
12.1.1	BundleViews Klassenreferenz	975
12.1.2	Configuration Klassenreferenz	976
12.1.3	Activator Klassenreferenz	977
12.1.4	Activator.HttpServiceTracker Klassenreferenz	979
12.1.5	Activator.ServletContextListenerServletAdaptor Klassenreferenz . . .	980
12.2	Implementierung	982
12.2.1	ExampleBean Klassenreferenz	982
12.2.2	MenuConstants Schnittstellenreferenz	986
12.2.3	MenuContributor Klassenreferenz	986
12.2.4	DataObjectScrollerTag Klassenreferenz	987
12.2.5	DataObjectTableTag Klassenreferenz	988

1 de.offis.pg.eam.core

1.1 BundleConfiguration

1.1.1 BundleViews Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::module::api::IModuleView.

1.1.1.1 Ausführliche Beschreibung

Diese Klasse schreibt die Views des Kerns in die Datenbank.

Autor:

Jens Henkel, Roland Koppe

Definiert in Zeile 29 der Datei BundleViews.java.

1.2 auth

1.2.1 IAuthNames Schnittstellenreferenz

1.2.1.1 Ausführliche Beschreibung

AUTO_CREATED INTERFACE FOR de.offis.pg.eam.core.auth.api.AuthNames

Definiert in Zeile 13 der Datei IAuthNames.java.

1.2.2 IDalBundleManager Schnittstellenreferenz

1.2.2.1 Ausführliche Beschreibung

AUTO_CREATED INTERFACE FOR de.offis.pg.eam.core.auth.api.DalBundleManager

Definiert in Zeile 22 der Datei IDalBundleManager.java.

1.2.3 IDalGroup Schnittstellenreferenz

1.2.3.1 Ausführliche Beschreibung

AUTO-CREATED INTERFACE FOR de.offis.pg.eam.core.auth.api.DalGroup
Definiert in Zeile 18 der Datei IDalGroup.java.

1.2.4 IDalRole Schnittstellenreferenz

1.2.4.1 Ausführliche Beschreibung

AUTO-CREATED INTERFACE FOR de.offis.pg.eam.core.auth.api.DalRole
Definiert in Zeile 24 der Datei IDalRole.java.

1.2.5 IDalUser Schnittstellenreferenz

1.2.5.1 Ausführliche Beschreibung

AUTO-CREATED INTERFACE FOR de.offis.pg.eam.core.auth.api.DalUser
Definiert in Zeile 18 der Datei IDalUser.java.

1.2.6 IMetamodelAuth Schnittstellenreferenz

1.2.6.1 Ausführliche Beschreibung

AUTO-CREATED INTERFACE FOR de.offis.pg.eam.core.auth.api.MetamodelAuth
Definiert in Zeile 13 der Datei IMetamodelAuth.java.

1.2.7 IUserInformation Schnittstellenreferenz

1.2.7.1 Ausführliche Beschreibung

AUTO-CREATED INTERFACE FOR de.offis.pg.eam.core.auth.api.UserInformation
Definiert in Zeile 14 der Datei IUserInformation.java.

1.2.8 AuthNames Klassenreferenz

1.2.8.1 Ausführliche Beschreibung

Sammlung der Rechte für das EAM-Tool.

Autor:

Roland Koppe

Definiert in Zeile 24 der Datei AuthNames.java.

Öffentliche, statische Methoden

- static String getAuthNameFor (Object object)
- static ArrayList< Right > getRightsById (long id)

1.2.8.2 Dokumentation der Elementfunktionen**static String getAuthNameFor (Object object) [static]**

Liefert den Namen eines Rechts für das angegebene Objekt. Gültige Objekte sind:

- im Metamodell
 - Category
 - EAMObject
 - EAMRelation
 - EAMAttribute
 - EAMObjectType
 - EAMRelationType
 - EAMAttributeType
- für Module
 - Module
 - Function
 - Action
 - MenuItem

Parameter:

object Ein beliebiges Objekt aus dem EAM-Tool, welches Rechte haben kann.

Rückgabe:

String Name des Rechts, sonst null

Definiert in Zeile 66 der Datei AuthNames.java.

static ArrayList<Right> getRightsById (long id) [static]

Liefert die zur angegebenen ID gehörenden Rechte der ENUM. Eine ID kann sich dabei aus mehreren IDs zusammensetzen, die aufaddiert alle Rechte beschreiben.

Parameter:

id Long ID des Rechtes

Rückgabe:

ArrayList<Right> Liste von Rechten, sollte die angegebene ID nicht korrekt sein, ist die Liste leer

Definiert in Zeile 135 der Datei AuthNames.java.

1.2.9 Dal Klassenreferenz

1.2.9.1 Ausführliche Beschreibung

Die Klasse Dal stellt die Zugriffe auf die Datenbank zur Verfügung.

Autor:

Christian R.

Definiert in Zeile 22 der Datei Dal.java.

Öffentliche, statische Methoden

- static ArrayList< String > getColumnNames (String table) throws DatabaseException
- static Object getSingleValues (String table, int id) throws SQLException, DatabaseException
- static boolean reloadUserRightsForRole (long roleId)
- static void removeFromTable (String table, int id) throws DatabaseException
- static void removeManyFromAssignmentTable (String table, String attribut, int id) throws DatabaseException
- static void removeOneFromAssignmentTable (String table, int id1, int id2) throws DatabaseException
- static void updateTable (String table, int id, ArrayList< String > updateValues) throws DatabaseException

1.2.9.2 Dokumentation der Elementfunktionen

static ArrayList<String> getColumnNames (String table) throws DatabaseException
[static]

Hole die Spalten Namen einer Tabelle

Parameter:

table Tabelle deren Spalten man haben möchte

Rückgabe:

Spalten Namen

Ausnahmebehandlung:**SQLException**

Definiert in Zeile 34 der Datei Dal.java.

Wird benutzt von Dal.removeOneFromAssignmentTable() und Dal.updateTable().

static Object getSingleValues (String table, int id) throws SQLException, DatabaseException [static]

Liefert nur eine einzige Zeile zurueck

Parameter:

table,entweder Role, Group oder User

id,die Id des Objekts

Rückgabe:

Object

Ausnahmebehandlung:**SQLException,DatabaseException**

Definiert in Zeile 69 der Datei Dal.java.

static boolean reloadUserRightsForRole (long roleId) [static]

Lädt die Rechte im UserManager für Benutzer mit der angegebenen Rolle neu.

Autor:

Roland Koppe

Parameter:

roleId long Rolle, zu der die Rechte neu geladen werden sollen

Rückgabe:

true, wenn erfolgreich, sonst false

Definiert in Zeile 110 der Datei Dal.java.

static void removeFromTable (String table, int id) throws DatabaseException [static]

Einen Datensatz aus einer Tabelle entfernen

Parameter:

table Betroffene Tabelle

id Primärschlüssel des Datensatzes

Ausnahmebehandlung:

SQLException,DatabaseException

Definiert in Zeile 144 der Datei Dal.java.

**static void removeManyFromAssignmentTable (String table, String attribut, int id)
throws DatabaseException [static]**

Entfernt alle zu einer id gehörenden Daten aus einer Verknüpfungs Tabelle.

Parameter:

table,Betroffene Tabelle.

attribut,Spalte der Tabelle.

id,Primaerschluessel des Datensatzes.

Ausnahmebehandlung:

SQLException,DatabaseException

Definiert in Zeile 163 der Datei Dal.java.

**static void removeOneFromAssignmentTable (String table, int id1, int id2) throws
DatabaseException [static]**

Entfernt eine Zeile aus einer Verknuepfungstabelle

Parameter:

table,Tabelle wie groups_role

id1,bei der Tabelle groups_role: groups_id

id2,bei der Tabelle groups_role: role_id

Rückgabe:

boolean

Ausnahmebehandlung:

SQLException,DatabaseException

Definiert in Zeile 183 der Datei Dal.java.

Benutzt Dal.getColumnNames().

**static void updateTable (String table, int id, ArrayList< String > updateValues)
throws DatabaseException [static]**

Ueberrnimmt Aenderungen an den Datensatzen in die entsprechende Tabelle

Parameter:

table,String aus dal.Constants
id,die Id des Primaerschluessels
updateValues,die zu aendernden Werte

Rückgabe:

boolean, ob erfolgreich oder nicht

Ausnahmebehandlung:

SQLException,DatabaseException

Definiert in Zeile 204 der Datei Dal.java.

Benutzt Dal.getColumnNames().

1.2.10 DalBundleManager Klassenreferenz

1.2.10.1 Ausführliche Beschreibung

Klasse mit statischen Methoden, um die Zugriffe auf die Datenbank zu kapseln. Diese Klasse stellt ausschließlich Methoden zur Verfügung, die für die Konfiguration der Bundles benötigt werden.

Bei Methoden, in denen eine ID des Bundle-Objektes übergeben werden soll, setzt sich diese ID wie folgt zusammen: Die Equinox-Id des jeweiligen Bundles + _ + der Name des jeweiligen Objektes + _ + die interne ID des Objektes aus dem Bundle

Die ID der Bundle-Attribute setzt sich auf ähnliche Weise zusammen: Die Equinox-Id des jeweiligen Bundles + _ + der Name des jeweiligen Attributes + _ + die interne ID des Objektes aus dem Bundle, zu dem das Attribut gehört

Autor:

Jens Henkel

Definiert in Zeile 36 der Datei DalBundleManager.java.

Öffentliche, statische Methoden

- static void addToModRoleAuthRole (String bundleRoleId, int coreRoleId)
- static boolean checkBundleEntriesInObjectAttributeMatching (String bundleId) throws SQLException, DatabaseException
- static boolean checkBundleEntriesInObjectMatching (String bundleId) throws SQLException, DatabaseException
- static boolean checkBundleEntriesInRelationAttributeMatching (String bundleId) throws SQLException, DatabaseException

- static boolean checkCoreEntriesInObjectAttributeMatching (String coreId) throws SQLException, DatabaseException
- static boolean checkCoreEntriesInObjectMatching (String coreId) throws SQLException, DatabaseException
- static boolean checkCoreEntriesInRelationAttributeMatching (String coreId) throws SQLException, DatabaseException
- static boolean checkMatchedBundleRelationEntries (String bundleRelationId)
- static boolean checkMatchedCoreRelationEntries (long coreRelationId)
- static boolean checkObjectAttributeMatches (String bundleId, long coreId) throws SQLException, DatabaseException
- static void deleteBundleViews (int equinoxId)
- static void deleteBundleViews (String bundleName)
- static void deleteObjectAttributeMatches (String equinoxId)
- static boolean deleteObjectAttributeMatches (String bundleId, String coreId)
- static void deleteObjectMatches (String bundleName)
- static boolean deleteObjectMatches (String bundleId, String coreId)
- static void deleteRelationAttributeMatches (String equinoxId)
- static boolean deleteRelationAttributeMatches (String bundleRelationId, String coreRelationId)
- static void deleteRelationMatches (String equinoxId)
- static boolean deleteRelationMatches (String bundleRelationId, String coreRelationId)
- static LinkedList< Long > getAllObjectIdsByBundleId (long bundleEquinoxId)
- static LinkedList< Long > getAllRelationIdsByBundleId (long bundleEquinoxId)
- static HashMap< String, String > getAssignedBundleToCoreEamObject (long bundleId)
- static String getBundleIdFromDatabase (String identifier, boolean isName) throws SQLException
- static LinkedList< Long > getBundleObjectDependenciesEquinoxId (String bundleDbId)
- static LinkedList< String > getBundleObjectDependenciesName (String bundleDbId)
- static View getBundleView (String bundleId, String viewName)
- static LinkedList< View > getBundleViews (String identifier, boolean isName) throws DatabaseException, SQLException
- static int getCoreMethodId (Method method)
- static HashMap< Long, LinkedList< Long > > getCoreObjectAttributeIdsFromObjectAttributeAssignmentTable (int equinoxId)
- static HashMap< Long, LinkedList< Long > > getCoreObjectAttributeIdsFromObjectAttributeAssignmentTable (String

- bundleName)
- static Long getCoreObjectIdFromObjectMapping (String bundleObjectId)
- static LinkedList< Long > getCoreObjectIdsFromObjectAssignmentTable (int equinoxId)
- static LinkedList< Long > getCoreObjectIdsFromObjectAssignmentTable (String bundleName)
- static HashMap< Long, LinkedList< Long > > getCoreRelationAttributeIdsFromRelationAttributeAssignmentTable (int equinoxId)
- static HashMap< Long, LinkedList< Long > > getCoreRelationAttributeIdsFromRelationAttributeAssignmentTable (String bundleId)
- static Long getCoreRelationIdByBundleRelationId (String bundleRelationId)
- static LinkedList< Long > getCoreRelationIdsFromRelationAssignmentTable (int equinoxId)
- static LinkedList< Long > getCoreRelationIdsFromRelationAssignmentTable (String bundleId)
- static EAMObjectAttributeMatchingStore getObjectAttributeMatches (String bundleId, long coreId) throws SQLException, DatabaseException
- static LinkedList< EAMObjectMatchingStore > getObjectMatchingByBundleObjectId (String bundleObjectId) throws SQLException, DatabaseException
- static EAMRelationAttributeMatchingStore getRelationAttributeMatches (String bundleId, long coreId) throws SQLException, DatabaseException
- static EAMRelationMatchingStore getRelationMatchesByBundleRelationId (String bundleRelationId) throws DatabaseException, SQLException
- static LinkedList< Method > getViewModuleMethods (int viewId)
- static LinkedList< Long > getViewRoleMatchesByBundleViewId (String viewId) throws SQLException, DatabaseException
- static int insertView (String viewName, String description, String zachman, String bundleDBId)
- static int insertViewCoreMethodAssignment (int methodId, int viewId, String bundleDbId)
- static int insertViewModuleMethod (Method info, String bundleId, int viewId)
- static int insertViewModuleMethodAssignment (int methodId, int viewId, String bundleId)
- static boolean removeAssignmentFromAuthRoleView (String viewId)
- static boolean removeSingleAssignmentFromAuthRoleView (String modViewId, int coreRoleId)
- static boolean removeSingleAssignmentFromModRole (String modRoleId, int coreRoleId)
- static boolean saveBundleToCoreAttributeAssignment (String bundleAttributeId,

- long coreAttributeId, String bundleObjId, long coreObjId, int storedAttribute, String bundleDbId)
- static boolean saveBundleToCoreObjectAssignment (String bundleObjectId, String coreObjectId, int isStoredObj, long categoryId, String categoryName, String bundleDbId)
- static boolean saveBundleToCoreRelationAssignment (String bundleRelationId, String coreRelationId, int storedRelation, long categoryId, String categoryName, String bundleDbId)
- static boolean saveBundleToCoreRelationAttributeAssignment (String bundleAttributeId, long coreAttributeId, String bundleRelId, long coreRelId, int storedAttribute, String bundleDbId)
- static boolean saveModRoleToCoreRoleAssignment (String bundleRoleId, int coreRoleId, long equinoxId)
- static boolean saveModViewToCoreRole (String viewId, String coreRoleId, String bundleDbId)
- static void updateView (String viewName, String description, String zachman, int viewId)
- static void updateViewAsError (int viewId)
- static boolean viewCoreMethodAssignmentExists (int methodId, int viewId)

1.2.10.2 Dokumentation der Elementfunktionen

static void addToModRoleAuthRole (String bundleRoleId, int coreRoleId) [static]

RoleMatchingHandler

Fügt einen Eintrag in die Verknüpfungs-Tabelle der Rollen mit den übergebenen Parametern hinzu.

Parameter:

bundleRoleId Die eindeutige ID der Modulrolle.

coreRoleId Die eindeutige ID der Kernrolle.

Definiert in Zeile 1006 der Datei DalBundleManager.java.

static boolean checkBundleEntriesInObjectAttributeMatching (String bundleId) throws SQLException, DatabaseException [static]

AttributeMatchingHandler

Überprüft, ob in der Mapping-Tabelle der Objekt-Attribute Einträge zu der übergebenen ID des Bundle-Attributes vorhanden sind.

Parameter:

bundleId Die eindeutige ID des Bundle-Attributes.

Rückgabe:

True, falls es Einträge zu der ID des Bundle-Attributes gibt, sonst false.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 1197 der Datei DalBundleManager.java.

static boolean checkBundleEntriesInObjectMatching (String bundleId) throws SQLException, DatabaseException [static]

ObjectMatchingHandler, RelationMatchingHandler

Überprüft, ob zu der angegebenen ID eines EAM-Objektes des Bundles ein Eintrag in der Mapping-Tabelle der Objekte vorhanden ist.

Parameter:

bundleId Die ID des EAM-Objektes des Bundles als String. Zusammengesetzt aus:
Equinox-Id des Bundles_Objektname_interne ID des Objektes für das Bundle

Rückgabe:

True, falls es Einträge zur CoreId gibt, sonst false.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 1503 der Datei DalBundleManager.java.

static boolean checkBundleEntriesInRelationAttributeMatching (String bundleId) throws SQLException, DatabaseException [static]

RelationAttributeMatchingHandler

Ermittelt, ob in der Verknüpfungs-Tabelle der Relations-Attribute bereits Einträge zu der übergebenen ID eines Relations-Attributes eines Moduls existieren.

Parameter:

bundleId Die eindeutige ID eines Relations-Attributes eines Moduls.

Rückgabe:

True, wenn Einträge existieren, sonst false.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 977 der Datei DalBundleManager.java.

static boolean checkCoreEntriesInObjectAttributeMatching (String coreId) throws SQLException, DatabaseException [static]

AttributeMatchingHandler

Überprüft, ob in der Mapping-Tabelle der Objekt-Attribute Einträge zu der übergebenen ID des Kernattributs vorhanden sind.

Parameter:

coreId Die eindeutige ID des Kernattributs.

Rückgabe:

True, falls es Einträge zu der ID des Kernattributs gibt, sonst false.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 1167 der Datei DalBundleManager.java.

static boolean checkCoreEntriesInObjectMatching (String coreId) throws SQLException, DatabaseException [static]

CoreBundleObjectMatchingHandler

Überprüft, ob zu der angegebenen ID eines EAM-Objektes des Kerns ein Eintrag in der Mapping-Tabelle der Objekte vorhanden ist.

Parameter:

coreId Die ID des EAM-Objektes des Kerns als String.

Rückgabe:

true, falls es Einträge zur CoreId gibt, sonst false.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 1576 der Datei DalBundleManager.java.

static boolean checkCoreEntriesInRelationAttributeMatching (String coreId) throws SQLException, DatabaseException [static]

RealtionAttributeMatchinghandler

Ermittelt, ob in der Verknüpfungs-Tabelle der Relations-Attribute bereits Einträge zu der übergebenen ID eines Relations-Attributes des Kerns existieren.

Parameter:

coreId Die eindeutige ID des Relations-Attributes des Kerns

Rückgabe:

True, wenn Einträge vorhanden sind, sonst false.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 907 der Datei DalBundleManager.java.

static boolean checkMatchedBundleRelationEntries (String bundleRelationId)
[static]

RelationMatchingHandler

Überprüft, ob zu der übergebenen ID einer Bundle-Relation ein Eintrag in der Mapping-Tabelle der Relationen vorhanden ist.

Parameter:

bundleRelationId Die eindeutige ID einer Bundle-Relation.

Rückgabe:

True, wenn es zu der ID Einträge gibt, sonst false.

Definiert in Zeile 737 der Datei DalBundleManager.java.

static boolean checkMatchedCoreRelationEntries (long coreRelationId) [static]

RelationMatchingHandler

Überprüft, ob zu der übergebenen ID einer Kern-Relation ein Eintrag in der Mapping-Tabelle der Relationen vorhanden ist.

Parameter:

coreRelationId Die eindeutige ID einer Kern-Relation.

Rückgabe:

True, wenn es zu der ID Einträge gibt, sonst false.

Definiert in Zeile 701 der Datei DalBundleManager.java.

static boolean checkObjectAttributeMatches (String bundleId, long coreId) throws SQLException, DatabaseException [static]

AttributeMatchingHandler

Überprüft, ob in der Mapping-Tabelle der Objekt-Attribute Einträge zu der Id des Kern- und des Bundle-Objektes vorhanden sind.

Parameter:

bundleId Die eindeutige ID des Bundle-Objektes.

coreId Die eindeutige ID des Kern-Objektes.

Rückgabe:

True, wenn ein Eintrag zu beiden IDs vorhanden ist, sonst false.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 1229 der Datei DalBundleManager.java.

static void deleteBundleViews (int equinoxId) [static]

BundleManager

Entfernt alle Views aus der DB, die zum Bundle mit der übergebenen Equinox-ID gehören

Parameter:

equinoxId Die Equinox-ID des Bundles.

Definiert in Zeile 2598 der Datei DalBundleManager.java.

Benutzt DalBundleManager.getIdFromDatabase().

static void deleteBundleViews (String bundleName) [static]

BundleManager

Entfernt alle Views aus der DB, die zum Bundle mit der übergebenen Equinox-ID gehören

Parameter:

bundleName Die Equinox-ID des Bundles.

Definiert in Zeile 2535 der Datei DalBundleManager.java.

Benutzt DalBundleManager.getIdFromDatabase().

static void deleteObjectAttributeMatches (String equinoxId) [static]

BundleManager

Entfernt alle Verknüpfungen von Objekt-Attributen, die zum Bundle mit der übergebenen Equinox-ID gehören.

Parameter:

equinoxId Die Equinox-ID des Bundles.

Definiert in Zeile 2363 der Datei DalBundleManager.java.

Benutzt DalBundleManager.getIdFromDatabase().

static boolean deleteObjectAttributeMatches (String bundleId, String coreId)

[static]

ObjectAttributeMatchingHandler

Entfernt aus der Mapping-Tabelle der Objekt-Attribute den Eintrag zu der übergebenen Bundle-Attribut-Id und der Kern-Attribut-Id.

Parameter:

bundleId Die eindeutige Bundle-Attribut-Id.

coreId Die eindeutige Kern-Attribut-Id.

Rückgabe:

True, wenn der Eintrag gelöscht werden konnte, sonst false.

Definiert in Zeile 1371 der Datei DalBundleManager.java.

static void deleteObjectMatches (String bundleName) [static]

CoreBundleObjectMatchingHandler

Entfernt alle Objekt-Verknüpfungen, die zum Bundle mit der übergebenen Equinox-ID gehören.

Parameter:

bundleName Die Equinox-ID des Bundles.

Definiert in Zeile 2333 der Datei DalBundleManager.java.

Benutzt DalBundleManager.getIdFromDatabase().

static boolean deleteObjectMatches (String bundleId, String coreId) [static]

ObjectMatchingHandler

Entfernt aus der Mapping-Tabelle der Objekte den Eintrag, der zu den übergebenen IDs passt.

Parameter:

bundleId Die eindeutige ID des Bundle-Objektes.

coreId Die eindeutige ID des Kern-Objektes.

Rückgabe:

True, wenn der Eintrag gelöscht werden konnte, sonst false.

Definiert in Zeile 1405 der Datei DalBundleManager.java.

static void deleteRelationAttributeMatches (String equinoxId) [static]

BundleManager

Entfernt alle Verknüpfungen von Relations-Attributen, die zum Bundle mit der übergebenen Equinox-ID gehören.

Parameter:

equinoxId Die Equinox-ID des Bundles.

Definiert in Zeile 2422 der Datei DalBundleManager.java.

Benutzt DalBundleManager.getBundleIdFromDatabase().

static boolean deleteRelationAttributeMatches (String bundleRelationId, String coreRelationId) [static]

RelationAttributeMatchingHandler

Löscht einen Eintrag aus der Verknüpfungstabelle der Relationsattribute, der zu den übergebenen IDs der Kern- und Bundle-Relationsattribute passt.

Parameter:

bundleRelationId Die eindeutige ID des Bundle-Relationsattributes.

coreRelationId Die eindeutige ID des Kern-Relationsattributes.

Rückgabe:

True, wenn der Eintrag gelöscht werden konnte, sonst false.

Definiert in Zeile 872 der Datei DalBundleManager.java.

static void deleteRelationMatches (String equinoxId) [static]

BundleManager

Entfernt alle Relations-Verknüpfungen, die zum Bundle mit der übergebenen Equinox-ID gehören.

Parameter:

equinoxId Die Equinox-ID des Bundles.

Definiert in Zeile 2392 der Datei DalBundleManager.java.

Benutzt DalBundleManager.getBundleIdFromDatabase().

static boolean deleteRelationMatches (String bundleRelationId, String coreRelationId) [static]

RelationMatchingHandler

Löscht aus der Verknüpfungs-Tabelle der Relationen einen passenden Eintrag zu den übergebenen Parametern.

Parameter:

bundleRelationId Die eindeutige ID der Bundle-Relation.

coreRelationId Die eindeutige ID der Kern-Relation.

Rückgabe:

True, wenn der Eintrag gelöscht werden konnte, sonst false.

Definiert in Zeile 816 der Datei DalBundleManager.java.

static LinkedList<Long> getAllObjectIdsByBundleId (long bundleEquinoxId) [static]

ModuleToCoreObjectWrapper

Ermittelt zu der übergebenen Bundle-Equinox-Id eine Liste aller EAMObjekte des Kerns, die mit EAMObjekten dieses Bundles verbunden sind.

Parameter:

bundleEquinoxId Die für das Bundle durch Equinox vergebene ID.

Rückgabe:

Eine Liste aller verknüpfter Objekte für dieses Bundle oder null im Fehlerfall.

Definiert in Zeile 1949 der Datei DalBundleManager.java.

static LinkedList<Long> getAllRelationIdsByBundleId (long bundleEquinoxId) [static]

ModuleToCoreObjectWrapper

Ermittelt zu der übergebenen Bundle-Equinox-Id eine Liste aller EAMRelationen des Kerns, die mit EAMRelationen dieses Bundles verbunden sind.

Parameter:

bundleEquinoxId Die für das Bundle durch Equinox vergebene ID.

Rückgabe:

Eine Liste aller verknüpfter Objekte für dieses Bundle oder null im Fehlerfall.

Definiert in Zeile 1991 der Datei DalBundleManager.java.

static HashMap<String, String> getAssignedBundleToCoreEamObject (long bundleId)
[static]

TestBundleMatching

Parameter:

bundleId Die eindeutige ID eines Bundles, die von Equinox vergeben wird.

Rückgabe:

Eine HashMap, in der die Namen der EAM-Objekte des Kerns und der Module abgelegt werden. Die IDs der Modul-Objekte sind dabei die Schlüssel und die Namen der Kern-Objekte die Values.

Definiert in Zeile 1032 der Datei DalBundleManager.java.

static String getBundleIdFromDatabase (String identifier, boolean isName) throws SQLException [static]

Liefert zu dem übergebenen Identifier (Name oder Equinox-Id) die durch die DB vergebene ID des Bundles.

Parameter:

identifier

isName True, wenn es sich bei identifier um den Namen des Bundles handelt. False, wenn es die Equinox-ID des Bundles ist.

Rückgabe:

Die ID des Bundles, die durch die DB vergeben wurde.

Ausnahmebehandlung:

SQLException

Definiert in Zeile 617 der Datei DalBundleManager.java.

Wird benutzt von DalBundleManager.deleteBundleViews(),
DalBundleManager.deleteObjectAttributeMatches(),
DalBundleManager.deleteObjectMatches(),
DalBundleManager.deleteRelationAttributeMatches(),
DalBundleManager.deleteRelationMatches(),
DalBundleManager.getCoreObjectAttributeIdsFromObjectAttributeAssignmentTable(),
DalBundleManager.getCoreObjectIdsFromObjectAssignmentTable(),
DalBundleManager.getCoreRelationAttributeIdsFromRelationAttributeAssignmentTable()
und DalBundleManager.getCoreRelationIdsFromRelationAssignmentTable().

static LinkedList<Long> getBundleObjectDependenciesEquinoxId (String bundleDbId)
[static]

Liefert zu der übergebenen BundleDbId eine Liste von Equinox-IDs anderer Bundles, die in irgendeiner Form von dem Bundle, das zur übergebenen ID gehört, abhängig ist.

Parameter:

bundleDbId Die durch die DB vergebene ID eines Bundles.

Rückgabe:

Eine Liste von Bundle-Namen, die von diesem Bundle abhängig sind.

Definiert in Zeile 45 der Datei DalBundleManager.java.

static LinkedList<String> getBundleObjectDependenciesName (String bundleDbId)
[static]

Liefert zu der übergebenen BundleDbId eine Liste von Equinox-IDs anderer Bundles, die in irgendeiner Form von dem Bundle, das zur übergebenen ID gehört, abhängig ist.

Parameter:

bundleDbId Die durch die DB vergebene ID eines Bundles.

Rückgabe:

Eine Liste von Bundle-Namen, die von diesem Bundle abhängig sind.

Definiert in Zeile 127 der Datei DalBundleManager.java.

static View getBundleView (String bundleId, String viewName) [static]

ModuleViewAdopter

Ermittelt zu der ID eines Bundles und dem Namen der View alle weiteren Informationen der View.

Parameter:

bundleId Die ID eines Bundles.

viewName Der Name einer View.

Rückgabe:

Ein zum Namen der View und der ID des Bundles gehörendes View-Objekt mit allen zusätzlichen Informationen der View.

Definiert in Zeile 2035 der Datei DalBundleManager.java.

static LinkedList<View> getBundleViews (String identifier, boolean isName) throws DatabaseException, SQLException [static]

BundleManager

Ermittelt zum übergebenen Identifier eine Liste der Views. Als identifier dient entweder der Name des Bundles oder dessen Equinox-ID.

Parameter:

identifizier Der Name des Bundles oder dessen Equinox-ID.

isName True, wenn es sich bei Identifier um den Namen des Bundles handelt. False, falls es die Equinox-Id ist.

Rückgabe:

Eine Liste der zum Bundle gehörenden Views.

Ausnahmebehandlung:

DatabaseException

SQLException

Definiert in Zeile 2254 der Datei DalBundleManager.java.

static int getCoreMethodId (Method method) [static]

ModueViewAdopter

Ermittelt zu dem übergebenen Kern-Method-Objekt die ID aus der Datenbank.

Parameter:

method Die Kern-Methode, zu der die ID ermittelt werden soll.

Rückgabe:

Die ID der Kernmethode oder -1 im Fehlerfall.

Definiert in Zeile 2664 der Datei DalBundleManager.java.

static HashMap<Long, LinkedList<Long> >

getCoreObjectAttributeIdsFromObjectAttributeAssignmentTable (int equinoxId)
[static]

BundleManager

Ermittelt zu der übergebenen Equinox-Id eines Bundles eine HashMap mit Ids von Attributen und Objekten. Als Key wird die ID eines Objektes und als Value eine LinkedList der Attribute dieses Objektes verwendet. Es werden nur die Attribute betrachtet, die ein StoredAttribute sind.

Parameter:

bundleName Die Equinox-Id eines Bunndles.

Rückgabe:

HashMap mit den gesetzten Werten.

Definiert in Zeile 358 der Datei DalBundleManager.java.

Benutzt DalBundleManager.getBundleIdFromDatabase().

```
static HashMap<Long, LinkedList<Long> >  
getCoreObjectAttributeIdsFromObjectAttributeAssignmentTable (String bundleName)  
[static]
```

BundleManager

Ermittelt zu der übergebenen Equinox-Id eines Bundles eine HashMap mit Ids von Attributen und Objekten. Als Key wird die ID eines Objektes und als Value eine LinkedList der Attribute dieses Objektes verwendet. Es werden nur die Attribute betrachtet, die ein StoredAttribute sind.

Parameter:

bundleName Die Equinox-Id eines Bunndles.

Rückgabe:

HashMap mit den gesetzten Werten.

Definiert in Zeile 301 der Datei DalBundleManager.java.

Benutzt DalBundleManager.getBundleIdFromDatabase().

```
static Long getCoreObjectIdFromObjectMapping (String bundleObjectId) [static]
```

RelationMatchingHandler

Ermittelt eine einzelne ID eines Kern-Objektes aus der Verknüpfungstabelle der Objekte.

Parameter:

bundleObjectId Die eindeutige ID eines Modul-Objektes.

Rückgabe:

Die Id des Kern-Objektes oder -1 im Fehlerfall.

Definiert in Zeile 1603 der Datei DalBundleManager.java.

```
static LinkedList<Long> getCoreObjectIdsFromObjectAssignmentTable (int  
equinoxId) [static]
```

BundleManager

Ermittelt zu der übergebenen Equinox-Id eines Bundles eine Liste von IDs der Kernobjekte, die mit einem EAM-Objekt des Bundles verknüpft wurden. Dabei handelt es sich nur um die IDs der Objekte, die als Objekte des Bundles in die DB übertragen und gespeichert wurden.

Parameter:

equinoxId Die Equinox-Id eines Bunndles.

Rückgabe:

Eine Liste von IDs der verknüpften Objekte.

Definiert in Zeile 257 der Datei DalBundleManager.java.

Benutzt DalBundleManager.getBundleIdFromDatabase().

static LinkedList<Long> getCoreObjectIdsFromObjectAssignmentTable (String bundleName) [static]

BundleManager

Ermittelt zu der übergebenen Equinox-Id eines Bundles eine Liste von IDs der Kernobjekte, die mit einem EAM-Objekt des Bundles verknüpft wurden. Dabei handelt es sich nur um die IDs der Objekte, die als Objekte des Bundles in die DB übertragen und gespeichert wurden.

Parameter:

bundleName Die Equinox-Id eines Bunndles.

Rückgabe:

Eine Liste von IDs der verknüpften Objekte.

Definiert in Zeile 212 der Datei DalBundleManager.java.

Benutzt DalBundleManager.getBundleIdFromDatabase().

static HashMap<Long, LinkedList<Long> > getCoreRelationAttributeIdsFromRelationAttributeAssignmentTable (int equinoxId) [static]

BundleManager

Ermittelt zu der übergebenen Equinox-Id eines Bundles eine HashMap mit Ids von Attributen und Relationen. Als Key wird die ID einer Relation und als Value eine LinkedList der Attribute dieser Relation verwendet. Es werden nur die Attribute betrachtet, die ein StoredAttribute sind.

Parameter:

bundleId Die Equinox-Id eines Bunndles.

Rückgabe:

HashMap mit den gesetzten Werten.

Definiert in Zeile 560 der Datei DalBundleManager.java.

Benutzt DalBundleManager.getBundleIdFromDatabase().

static HashMap<Long, LinkedList<Long> > getCoreRelationAttributeIdsFromRelationAttributeAssignmentTable (String bundleId)

[static]

BundleManager

Ermittelt zu der übergebenen Equinox-Id eines Bundles eine HashMap mit Ids von Attributen und Relationen. Als Key wird die ID einer Relation und als Value eine LinkedList der Attribute dieser Relation verwendet. Es werden nur die Attribute betrachtet, die ein StoredAttribute sind.

Parameter:

bundleId Die Equinox-Id eines Bunndles.

Rückgabe:

HashMap mit den gesetzten Werten.

Definiert in Zeile 503 der Datei DalBundleManager.java.

Benutzt DalBundleManager.getBundleIdFromDatabase().

static Long getCoreRelationIdByBundleRelationId (String bundleRelationId) [static]

ModuleToCoreObjectWrapper

Ermittelt eine einzelne ID einer Kern-Relation aus der Verknüpfungstabelle der Relationen.

Parameter:

bundleRelationId Die eindeutige ID einer Modul-Relation.

Rückgabe:

Die Id des Kern-Objektes oder -1 im Fehlerfall.

Definiert in Zeile 1909 der Datei DalBundleManager.java.

static LinkedList<Long> getCoreRelationIdsFromRelationAssignmentTable (int equinoxId) [static]

BundleManager

Ermittelt zu der übergebenen Equinox-Id eines Bundles eine Liste von IDs der Kernrelationen, die mit einer EAM-Relation des Bundles verknüpft wurden. Dabei handelt es sich nur um die IDs der Relationen, die als Relationen des Bundles in die DB übertragen und gespeichert wurden.

Parameter:

bundleId Die Equinox-Id eines Bunndles.

Rückgabe:

Eine Liste von IDs der verknüpften Relationen.

Definiert in Zeile 459 der Datei DalBundleManager.java.

Benutzt DalBundleManager.getBundleIdFromDatabase().

static LinkedList<Long> getCoreRelationIdsFromRelationAssignmentTable (String bundleId) [static]

BundleManager

Ermittelt zu der übergebenen Equinox-Id eines Bundles eine Liste von IDs der Kernrelationen, die mit einer EAM-Relation des Bundles verknüpft wurden. Dabei handelt es sich nur um die IDs der Relationen, die als Relationen des Bundles in die DB übertragen und gespeichert wurden.

Parameter:

bundleId Die Equinox-Id eines Bunndles.

Rückgabe:

Eine Liste von IDs der verknüpften Relationen.

Definiert in Zeile 414 der Datei DalBundleManager.java.

Benutzt DalBundleManager.getBundleIdFromDatabase().

static EAMObjectAttributeMatchingStore getObjectAttributeMatches (String bundleId, long coreId) throws SQLException, DatabaseException [static]

AttributeMatchingHandler

Überprüft, ob in der Mapping-Tabelle der Objekt-Attribute Einträge zu der Id des Kern- und des Bundle-Objektes vorhanden sind.

Parameter:

bundleId Die eindeutige ID des Bundle-Objektes.

coreId Die eindeutige ID des Kern-Objektes.

Rückgabe:

True, wenn ein Eintrag zu beiden IDs vorhanden ist, sonst false.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 1262 der Datei DalBundleManager.java.

Benutzt EAMObjectAttributeMatchingStore.setBundleAttributeId(), EAMObjectAttributeMatchingStore.setCoreAttributeId() und EAMObjectAttributeMatchingStore.setStoredAttribute().

static LinkedList<EAMObjectMatchingStore> getObjectMatchingByBundleObjectId (String bundleObjectId) throws SQLException,

DatabaseException [static]

ObjectMatchingHandler

Ermittelt aus der Mapping-Tabelle der EAM-Objekte die zur übergebenen BundleId gehörenden Matching-Einträge.

Parameter:

bundleObjectId Die eindeutige ID des Bundle-Objektes.

Rückgabe:

Eine Liste von MappingStores, die die in der Tabelle befindlichen Informationen kapselt. Die ObjectMappingStores könnten noch umgestellt werden auf EAMObjectMatchingStores, da diese im Grunde die gleiche Aufgabe erledigen.

Ausnahmebehandlung:**SQLException****DatabaseException**

Definiert in Zeile 1297 der Datei DalBundleManager.java.

Benutzt EAMObjectMatchingStore.setBundleObjectId(),
EAMObjectMatchingStore.setCoreObjectId(),
EAMObjectMatchingStore.setMetamodelId(),
EAMObjectMatchingStore.setMetamodelName() und
EAMObjectMatchingStore.setStoredObject().

static EAMRelationAttributeMatchingStore getRelationAttributeMatches (String bundleId, long coreId) throws SQLException, DatabaseException [static]

RelationAttributeMatchingHandler

Ermittelt, ob es zu den IDs des Kern-Relationsattributes und des Modul-Relationsattributes einen Eintrag in der Verknüpfungstabelle der Relationsattribute gibt. Sollte es diesen Eintrag geben, wird ein EAMRelationAttributeMatchingStore angelegt und zurückgegeben.

Parameter:

bundleId Die eindeutige ID des Bundle-Relationsattributes.

coreId Die einseitige ID des Kern-Relationsattributes.

Rückgabe:

Einen teilweise gefüllten EAMRelationAttributeMatchingStore. Lediglich der Name des Modulattributes und der des Kernattributes müssen noch hinzugefügt werden. Liefert null wenn es keinen passenden Eintrag gibt.

Ausnahmebehandlung:**SQLException**

DatabaseException

Definiert in Zeile 941 der Datei DalBundleManager.java.

Benutzt EAMRelationAttributeMatchingStore.setBundleRelationAttributeId(),
EAMRelationAttributeMatchingStore.setCoreRelationAttributeId() und
EAMRelationAttributeMatchingStore.setStoredAttribute().

**static EAMRelationMatchingStore getRelationMatchesByBundleRelationId (String
bundleRelationId) throws DatabaseException, SQLException [static]**

RelationMatchingHandler

Ermittelt für die übergebene ID einer Bundle-Relation, ob zu dieser ein Eintrag in der Verknüpfungstabelle der Relationen vorhanden ist. Ist dies der Fall, wird ein EAMRelationMatchingStore erstellt und mit den aus der Verknüpfungstabelle entnommenen Daten gefüllt.

Parameter:

bundleRelationId Die eindeutige ID der Bundle-Relation.

Rückgabe:

Der erstellte EAMRelationMatchingStore, wenn zu der übergebenen ID ein Eintrag gefunden wurde, sonst null.

Ausnahmebehandlung:

DatabaseException

SQLException

Definiert in Zeile 666 der Datei DalBundleManager.java.

Benutzt EAMRelationMatchingStore.setBundleRelationId(),
EAMRelationMatchingStore.setCategoryId(),
EAMRelationMatchingStore.setCategoryName(),
EAMRelationMatchingStore.setCoreRelationId() und
EAMRelationMatchingStore.setStoredRelation().

static LinkedList<Method> getViewModuleMethods (int viewId) [static]

Parameter:

viewId

Rückgabe:

Definiert in Zeile 2799 der Datei DalBundleManager.java.

```
static LinkedList<Long> getViewRoleMatchesByBundleViewId (String viewId) throws  
SQLException, DatabaseException [static]
```

CoreBundleRoleMatchingHandler

Ermittelt zu der übergebenen ID einer View, ob zu dieser View bereits Verknüpfungen mit Rollen des Kerns existieren.

Parameter:

viewId Die ID einer View eines Bundles.

Rückgabe:

Eine Liste der mit einer View verknüpften Rollen. Diese ist leer, falls es noch keine Verknüpfungen gibt.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 1657 der Datei DalBundleManager.java.

```
static int insertView (String viewName, String description, String zachman, String  
bundleDBId) [static]
```

ModuleViewAdopter

Fügt in die Tabelle der Views einen neuen Eintrag mit einer neuen View hinzu.

Parameter:

viewName Der Name der View.

description Eine Beschreibung der View.

zachman Der Zachman-Standard der View.

bundleDBId Die ID des Bundles, die durch die DB vergeben wurde.

Rückgabe:

Die ID der View, die durch die DB vergeben wurde, oder -1 im Fehlerfall.

Definiert in Zeile 2119 der Datei DalBundleManager.java.

```
static int insertViewCoreMethodAssignment (int methodId, int viewId, String  
bundleDbId) [static]
```

ModuleViewAdopter

Fügt in der Verknüpfungstabelle der Kern-Methoden und den Views einen Eintrag ein.

Parameter:

methodId Die ID der Kern-Methode, die durch die DB vergeben wurde.

viewId Die ID der View, die durch die PG vergeben wurde.

Rückgabe:

0, wenn der Eintrag eingefügt werden konnte, -1 im Fehlerfall.

Definiert in Zeile 2225 der Datei DalBundleManager.java.

static int insertViewModuleMethod (Method info, String bundleId, int viewId)
[static]

ModuleViewAdopter

Fügt in die Tabelle der Modul-Viewmethoden einen neuen Eintrag ein.

Parameter:

info Informationen über die Methode in einem Method-Objekt.

Rückgabe:

Die ID der Methode, die durch die DB vergeben wurde, oder -1 im Fehlerfall.

Definiert in Zeile 2153 der Datei DalBundleManager.java.

static int insertViewModuleMethodAssignment (int methodId, int viewId, String bundleId) [static]

ModuleViewAdopter

Fügt in der Verknüpfungstabelle der Methoden mit den Views einen Eintrag ein.

Parameter:

methodId Die ID der Methode, die durch die DB vergeben wurde.

viewId Die ID der View, die durch die DB vergeben wurde.

Rückgabe:

0, wenn der Eintrag eingefügt werden konnte, -1 im Fehlerfall.

Definiert in Zeile 2194 der Datei DalBundleManager.java.

static boolean removeAssignmentFromAuthRoleView (String viewId) [static]

CoreBundleRoleMatchingHandler

Entfernt alle Einträge aus der Verknüpfungs-Tabelle der Rollen, die zu der übergebenen ID einer Modul-Rolle gehört.

Parameter:

viewId Die eindeutige ID der Modulrolle.

Rückgabe:

True, wenn der Eintrag gelöscht werden konnte, sonst false.

Definiert in Zeile 1689 der Datei DalBundleManager.java.

static boolean removeSingleAssignmentFromAuthRoleView (String modViewId, int coreRoleId) [static]

CoreBundleRoleMatchingHandler

Entfernt einen einzelnen Eintrag aus der Verknüpfungs-Tabelle der Rollen, die zur übergebenen ID der Modul- und Kern-Rolle passt.

Parameter:

modViewId Die eindeutige ID der Modulrolle.

coreRoleId Die eindeutige ID der Kernrolle.

Rückgabe:

True, wenn der Eintrag gelöscht werden konnte, sonst false.

Definiert in Zeile 1751 der Datei DalBundleManager.java.

static boolean removeSingleAssignmentFromModRole (String modRoleId, int coreRoleId) [static]

CoreBundleRoleMatchingHandler

Entfernt einen einzelnen Eintrag aus der Verknüpfungs-Tabelle der Rollen, die zur übergebenen ID der Modul- und Kern-Rolle passt.

Parameter:

modRoleId Die eindeutige ID der Modulrolle.

coreRoleId Die eindeutige ID der Kernrolle.

Rückgabe:

True, wenn der Eintrag gelöscht werden konnte, sonst false.

Definiert in Zeile 1719 der Datei DalBundleManager.java.

static boolean saveBundleToCoreAttributeAssignment (String bundleAttributId, long coreAttributId, String bundleObjId, long coreObjId, int storedAttribute, String bundleDbId) [static]

AttributeMatchingHandler

Fügt der Mapping-Tabelle der Objekt-Attribute einen neuen Eintrag mit den übergebenen Informationen hinzu.

Parameter:

bundleAttributId Die eindeutige ID des Bundle-Attributes.

coreAttributId Die eindeutige ID des Kern-Attributes.

bundleObjId Die eindeutige ID des Bundle-Objektes.

coreObjId Die eindeutige ID des Kern-Objektes.

storedAttribute 0, falls es sich bei dem Eintrag um einen gemappten handelt. 1, wenn es sich um ein Attribut handelt, das einem bereits existierenden Objekt hinzugefügt werden soll.

Rückgabe:

True, wenn der Eintrag hinzugefügt werden konnte, sonst false.

Definiert in Zeile 1088 der Datei DalBundleManager.java.

static boolean saveBundleToCoreObjectAssignment (String bundleObjectId, String coreObjectId, int isStoredObj, long categoryId, String categoryName, String bundleDbId) [static]

ObjectMatchingHandler

Fügt der Mapping-Tabelle der Objekte einen neuen Eintrag hinzu.

Parameter:

bundleObjectId Die eindeutige ID eines Bundle-Objektes.

coreObjectId Die eindeutige ID eines Kern-Objektes.

isStoredObj 0, falls es sich bei dem Eintrag nur um ein gemapptes Objekt handelt. 1, wenn es sich um ein EAM-Objekt handelt, das den Kern-Objekten hinzugefügt wurde.

categoryId Die ID der Category, zu der das Stored Object hinzugefügt wurde.

categoryName Der Name der Category, zu der das Stored Object hinzugefügt wurde.

Rückgabe:

True, wenn ein neuer Eintrag hinzugefügt werden konnte, sonst false.

Definiert in Zeile 1462 der Datei DalBundleManager.java.

static boolean saveBundleToCoreRelationAssignment (String bundleRelationId, String coreRelationId, int storedRelation, long categoryId, String categoryName, String bundleDbId) [static]

RelationMatchingHandler

Parameter:

bundleRelationId Die eindeutige ID der Bundlerelation.

coreRelationId Die eindeutige ID der Kernrelation.

storedRelation 1, wenn es sich um eine Stored Relation handelt, die Relation also direkt in den Kerngespeichert wurde. 0 sonst.

categoryId Die eindeutige ID der Category, zu der die Relation hinzugefügt wurde oder null.

categoryName Der Name der Category, zu der die Relation hinzugefügt wurde oder null.

Rückgabe:

True, wenn der Eintrag erfolgreich angelegt werden konnte, sonst false.

Definiert in Zeile 778 der Datei DalBundleManager.java.

static boolean saveBundleToCoreRelationAttributeAssignment (String bundleAttributeld, long coreAttributeld, String bundleRelld, long coreRelld, int storedAttribute, String bundleDbID) [static]

RelationAttributeMatchingHandler

Fügt der Mapping-Tabelle der Relations-Attribute einen neuen Eintrag mit den übergebenen Informationen hinzu.

Parameter:

bundleAttributeld Die eindeutige ID des Bundle-Attributes.

coreAttributeld Die eindeutige ID des Kern-Attributes.

bundleRelld Die eindeutige ID des Bundle-Objektes.

coreRelld Die eindeutige ID des Kern-Objektes.

storedAttribute 0, falls es sich bei dem Eintrag um einen gemappten handelt. 1, wenn es sich um ein Attribut handelt, das einem bereits existierenden Objekt hinzugefügt werden soll.

Rückgabe:

True, wenn der Eintrag hinzugefügt werden konnte, sonst false.

Definiert in Zeile 1129 der Datei DalBundleManager.java.

static boolean saveModRoleToCoreRoleAssignment (String bundleRoleld, int coreRoleld, long equinoxId) [static]

CoreBundleRoleMatchingHandler

Speichert einen Eintrag in der Verknüpfungs-Tabelle der Rollen.

Parameter:

bundleRoleld Die eindeutige ID der Modul-Rolle.

coreRoleld Die eindeutige ID der Kern-Rolle.

Rückgabe:

True, wenn der Eintrag gespeichert werden konnte, sonst false.

Definiert in Zeile 1782 der Datei DalBundleManager.java.

static boolean saveModViewToCoreRole (String viewId, String coreRoleId, String bundleDbId) [static]

CoreBundleRoleMatchingHandler

Speichert einen Eintrag in der Verknüpfungs-Tabelle der Modul-Views mit den Kern-Rollen.

Parameter:

viewId Die eindeutige ID der Modul-View.

coreRoleId Die eindeutige ID der Kern-Rolle.

Rückgabe:

True, wenn der Eintrag gespeichert werden konnte, sonst false.

Definiert in Zeile 1846 der Datei DalBundleManager.java.

static void updateView (String viewName, String description, String zachman, int viewId) [static]

ModuleViewAdopter

Aktualisiert den Eintrag einer View in der DB.

Parameter:

viewName Der Name der View.

description Eine Beschreibung der View.

zachman Der Zachman-Standard der View.

viewId Die ID der View.

Definiert in Zeile 2734 der Datei DalBundleManager.java.

static void updateViewAsError (int viewId) [static]

ModuleViewAdopter

Markiert eine View in der DB als fehlerhaft, falls z.B. eine Kernmethode nicht ermittelt werden konnte.

Parameter:

viewId Die ID der View, die als fehlerhaft markiert werden soll.

Definiert in Zeile 2708 der Datei DalBundleManager.java.

static boolean viewCoreMethodAssignmentExists (int methodId, int viewId)
[static]

ModuleViewAdopter

Überprüft, ob es zu der übergebenen ID einer Kernmethode und der ID einer View bereits einen Eintrag in der Verknüpfungstabelle gibt.

Parameter:

methodId Die ID einer Kernmethode.

viewId Die ID einer View

Rückgabe:

True, falls bereits einen Eintrag existiert, sonst false.

Definiert in Zeile 2766 der Datei DalBundleManager.java.

1.2.11 DalGroup Klassenreferenz

1.2.11.1 Ausführliche Beschreibung

Die Klasse DalGroup stellt Methoden zum Bearbeiten und Lesen von Gruppen bereit
Implementiert das Interface IDalGroup aus dem Paket auth.api.

Autor:

Christian R.

Definiert in Zeile 23 der Datei DalGroup.java.

Öffentliche, statische Methoden

- static void assignRole (int groupId, int roleId) throws DatabaseException
- static void assignUser (int groupId, int userId) throws DatabaseException
- static void createGroup (String name, String description) throws DatabaseException
- static void deleteGroup (int groupId) throws DatabaseException, SQLException, NotDeletableException
- static ArrayList< Role > getAssignedRoles (int groupId) throws SQLException, DatabaseException
- static ArrayList< User > getAssignedUsers (int groupId) throws SQLException, DatabaseException
- static Integer getGroupIdByName (String groupName) throws SQLException, DatabaseException
- static ArrayList< Group > getGroups () throws DatabaseException, SQLException
- static Group getSingleGroup (int groupId) throws SQLException, DatabaseException

- static void removeAssignedRole (int groupId, int roleId) throws DatabaseException, SQLException
- static void removeAssignedUser (int groupId, int userId) throws DatabaseException, SQLException
- static void updateGroup (int groupId, String name, String description) throws DatabaseException, SQLException

1.2.11.2 Dokumentation der Elementfunktionen

static void assignRole (int groupId, int roleId) throws DatabaseException
[static]

Ordnet einer Gruppe eine Rolle zu.

Parameter:

groupId,die Id der Gruppe.

roleId,die Id der Rolle.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 215 der Datei DalGroup.java.

static void assignUser (int groupId, int userId) throws DatabaseException
[static]

Ordnet einer Gruppe einen Benutzer zu.

Parameter:

groupId,die Id der Gruppe.

userId,die Id des Benutzers.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 234 der Datei DalGroup.java.

static void createGroup (String name, String description) throws DatabaseException [static]

Erstellt eine Gruppe in der Datenbank.

Parameter:

name,Der Name der Gruppe.

description,Die dazugehoerige Beschreibung.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 57 der Datei DalGroup.java.

static void deleteGroup (int groupId) throws DatabaseException, SQLException, NotDeletableException [static]

Entfernt eine Gruppe und die Verknuepfungen mit Rollen und Benutzern aus der Datenbank.

Parameter:

groupId, die Id der Gruppe, die geloescht werden soll.

Ausnahmebehandlung:

SQLException

DatabaseException

SQLException

NotDeletableException

Definiert in Zeile 160 der Datei DalGroup.java.

Benutzt DalGroup.getSingleGroup().

static ArrayList<Role> getAssignedRoles (int groupId) throws SQLException, DatabaseException [static]

Liefert die Rollen, die der Gruppe zugewiesen sind.

Parameter:

groupId, die Id der Gruppe.

Rückgabe:

Die Liste der zugewiesenen Rollen mit Id, Name und Beschreibung.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 89 der Datei DalGroup.java.

static ArrayList<User> getAssignedUsers (int groupId) throws SQLException, DatabaseException [static]

Liefert die Benutzer, die dieser Gruppe angehören.

Parameter:

groupId, die Id, der entsprechenden Gruppe

Rückgabe:

eine Liste mit Benutzern mit Id, Vorname, Nachname, Passwort und Arbeitsplatz.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 128 der Datei DalGroup.java.

static Integer getGroupIdByName (String groupName) throws SQLException, DatabaseException [static]

Liefert die Id einer Gruppe anhand des Namens.

Parameter:

groupName

Rückgabe:

die Id der Gruppe.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 76 der Datei DalGroup.java.

static ArrayList<Group> getGroups () throws DatabaseException, SQLException [static]

Liefert alle Gruppen.

Rückgabe:

groups, die in der Datenbank gespeicherten Gruppen.

Ausnahmebehandlung:

DatabaseException

SQLException

Definiert in Zeile 31 der Datei DalGroup.java.

static Group getSingleGroup (int groupId) throws SQLException, DatabaseException [static]

Liefert eine einzelne Gruppe anhand der Id

Parameter:

groupId,die Id der gewünschten Gruppe.

Rückgabe:

Ein Gruppenobjekt.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 116 der Datei DalGroup.java.

Wird benutzt von DalGroup.deleteGroup().

static void removeAssignedRole (int groupId, int roleId) throws DatabaseException, SQLException [static]

Entfernt aus der Verknuepfungstabelle zwischen Gruppen und Rollen einen Eintrag.

Parameter:

groupId,die Id der Gruppe.

roleId,die Id der Rolle.

Ausnahmebehandlung:

SQLException

DatabaseException

SQLException

Definiert in Zeile 202 der Datei DalGroup.java.

static void removeAssignedUser (int groupId, int userId) throws DatabaseException, SQLException [static]

Entfernt aus der Verknuepfungstabelle zwischen Gruppen und Benutzern einen Eintrag.

Parameter:

groupId,die Id der Gruppe.

userId,die Id des Benutzers.

Ausnahmebehandlung:

SQLException

DatabaseException

SQLException

Definiert in Zeile 188 der Datei DalGroup.java.

static void updateGroup (int groupId, String name, String description) throws DatabaseException, SQLException [static]

Uebernimmt neue Werte fuer eine Gruppe.

Parameter:

id,die Id der Gruppe

updateValues,die neuen Daten.

Ausnahmebehandlung:

SQLException

DatabaseException

SQLException

Definiert in Zeile 255 der Datei DalGroup.java.

1.2.12 DalRole Klassenreferenz

1.2.12.1 Ausführliche Beschreibung

Die Klasse DalRole stellt Methoden zur Rollenbearbeitung bereit. Implementiert das Interface aus dem Paket IDalRole aus dem Paket auth.api.

Autor:

Christian R.

Definiert in Zeile 34 der Datei DalRole.java.

Öffentliche, statische Methoden

- static void assignFunc (long viewId, int roleId, long bundleId) throws DatabaseException
- static void assignGroup (int groupId, int roleId) throws DatabaseException, SQLException
- static void assignRight (int roleId, String name, String description, String right, int access, boolean metaType) throws DatabaseException, SQLException
- static void assignUser (int userId, int roleId) throws DatabaseException, SQLException
- static void createRole (String name, String description) throws DatabaseException
- static void deleteRole (int roleId) throws DatabaseException, SQLException, NotDeletableException
- static ArrayList< Group > getAssignedGroups (int roleId) throws SQLException, DatabaseException
- static ArrayList< AuthInfo > getAssignedRights (int roleId, boolean metaType) throws NumberFormatException, SQLException, DatabaseException

- static ArrayList< User > getAssignedUsers (int roleId) throws SQLException, DatabaseException
- static ArrayList< View > getAssignedViews (int roleId) throws DatabaseException, SQLException
- static ArrayList< Module > getBundleRights () throws DatabaseException, SQLException
- static Role getRoleById (Long id) throws SQLException, DatabaseException
- static Integer getRoleIdByName (String name) throws SQLException, DatabaseException
- static ArrayList< Role > getRoles () throws SQLException, DatabaseException
- static Role getSingleRole (int roleId) throws SQLException, DatabaseException
- static HashSet< Pair > getViewsByUser (long userId) throws DatabaseException, SQLException
- static void removeAssignedFunc (long viewId, int roleId) throws DatabaseException
- static void removeAssignedGroup (int groupId, int roleId) throws DatabaseException, SQLException
- static void removeAssignedUser (int userId, int roleId) throws DatabaseException, SQLException
- static void updateAccess (long id, String access, boolean metaRight) throws DatabaseException
- static void updateRole (int roleId, String name, String description) throws DatabaseException, SQLException

1.2.12.2 Dokumentation der Elementfunktionen

static void assignFunc (long viewId, int roleId, long bundleId) throws DatabaseException [static]

Weisst einer Rolle eine View zu.

Parameter:

funcId

roleId

Ausnahmebehandlung:

DatabaseException

Definiert in Zeile 847 der Datei DalRole.java.

static void assignGroup (int groupId, int roleId) throws DatabaseException, SQLException [static]

Ordnet einer Rolle eine Gruppe zu und laedt die Rechte der angemeldeten Nutzer neu, falls diese die Gruppe besitzen.

Parameter:

groupId, die Id der Gruppe.

roleId, die Id der Rolle.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 65 der Datei DalRole.java.

static void assignRight (int roleId, String name, String description, String right, int access, boolean metaType) throws DatabaseException, SQLException [static]

Ordnet einer Rolle ein Recht zu.

Parameter:

roleId, die Id der Rolle.

name, Name des Rechts.

description, Beschreibung des Rechts.

right, das Recht selbst.

access, Zugriffsart.

Ausnahmebehandlung:

SQLException

DatabaseException

SQLException

Definiert in Zeile 110 der Datei DalRole.java.

static void assignUser (int userId, int roleId) throws DatabaseException, SQLException [static]

Ordnet eine Rolle einem Benutzer zu und laedt dessen Rechte neu.

Parameter:

userId, die Id des Benutzers.

roleId, die Id der Rolle.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 86 der Datei DalRole.java.

static void createRole (String name, String description) throws DatabaseException
[static]

Erstellt eine neue Rolle in der Datenbank.

Parameter:

name,der Name der Rolle.

description,die Beschreibung der Rolle.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 44 der Datei DalRole.java.

static void deleteRole (int roleId) throws DatabaseException, SQLException, NotDeletableException [static]

Loescht eine Rolle mit allen Verknuepfungen aus der Datenbank.

Parameter:

roleId,die Id der Rolle.

Ausnahmebehandlung:

SQLException

DatabaseException

NotDeletableException,falls eine Rolle geloescht werden sollte, die nicht loeschbar sein soll.

Definiert in Zeile 356 der Datei DalRole.java.

Benutzt DalRole.getSingleRole().

static ArrayList<Group> getAssignedGroups (int roleId) throws SQLException, DatabaseException [static]

Liefert die der Rolle zugeordneten Gruppen.

Parameter:

roleId

Rückgabe:

eine Liste mit Gruppen, die der Rolle zugeordnet sind, mit Id, Name und Beschreibung.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 543 der Datei DalRole.java.

**static ArrayList<AuthInfo> getAssignedRights (int roleId, boolean metaType)
throws NumberFormatException, SQLException, DatabaseException [static]**

Liefert die der Rolle zugeordneten Rechte.

Parameter:

roleId, die Id der Rolle.

Rückgabe:

Die Liste der zugewiesenen Rechte.

Ausnahmebehandlung:

NumberFormatException

SQLException

DatabaseException

Definiert in Zeile 218 der Datei DalRole.java.

Benutzt AuthInfo.setAccess(), AuthInfo.setId() und AuthInfo.setRight().

**static ArrayList<User> getAssignedUsers (int roleId) throws SQLException,
DatabaseException [static]**

Liefert die der Rolle zugeordneten Benutzer

Parameter:

roleId

Rückgabe:

eine Liste mit Benutzern, die der Rolle zugeordnet sind, mit Id, Login und Arbeitsplatz.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 572 der Datei DalRole.java.

**static ArrayList<View> getAssignedViews (int roleId) throws DatabaseException,
SQLException [static]**

Erzeugt eine Liste mit Views von Bundles, welcher einer Rolle zugehoeren.

Parameter:

roleId, die Id der Rolle.

Rückgabe:

die Liste mit den Rechten.

Ausnahmebehandlung:

DatabaseException

SQLException

Definiert in Zeile 809 der Datei DalRole.java.

static ArrayList<Module> getBundleRights () throws DatabaseException, SQLException [static]

Erzeugt eine Liste mit Bundles und fuegt jedem Bundle die zugehoerigen Views hinzu. Ein View wird aber nicht hinzugefuegt, wenn dieser fehlerhaft ist. Siehe BundleKonfig.

Rückgabe:

die Bundles mit ihren Rechten

Ausnahmebehandlung:

DatabaseException

SQLException

Definiert in Zeile 763 der Datei DalRole.java.

static Role getRoleById (Long id) throws SQLException, DatabaseException [static]

Liefert eine Rolle anhand ihrer Id.

Parameter:

id, die Id der Rolle

Rückgabe:**Ausnahmebehandlung:**

SQLException

DatabaseException

Definiert in Zeile 607 der Datei DalRole.java.

static Integer getRoleIdByName (String name) throws SQLException, DatabaseException [static]

Liefert die Id einer Rolle anhand ihres Namens.

Parameter:

name,der Name der Rolle.

Rückgabe:

die Id der Rolle.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 635 der Datei DalRole.java.

static ArrayList<Role> getRoles () throws SQLException, DatabaseException
[static]

Liefert alle Rollen des Systems

Rückgabe:

alle Rollen, mit Id, Name und Beschreibung.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 647 der Datei DalRole.java.

static Role getSingleRole (int roleId) throws SQLException, DatabaseException
[static]

Liefert eine Rolle anhand ihrer Id.

Parameter:

roleId

Rückgabe:

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 676 der Datei DalRole.java.

Wird benutzt von DalRole.deleteRole().

static HashSet<Pair> getViewsByUser (long userId) throws DatabaseException, SQLException [static]

Erstellt eine Liste mit Views, über die ein Benutzer durch die ihm zugewiesenen Rollen und Gruppen verfügt.

Parameter:

userId, die Id des Benutzers

Rückgabe:

eine HashSet<String>, die Werte bestehen aus: BundleName + ViewName

Ausnahmebehandlung:

DatabaseException

SQLException

Definiert in Zeile 889 der Datei DalRole.java.

static void removeAssignedFunc (long viewId, int roleId) throws DatabaseException [static]

Entfernt eine Verknüpfung zwischen einer View und einer Rolle

Parameter:

viewId

roleId

Ausnahmebehandlung:

DatabaseException

Definiert in Zeile 867 der Datei DalRole.java.

static void removeAssignedGroup (int groupId, int roleId) throws DatabaseException, SQLException [static]

Entfernt die Verknüpfung zwischen einer Rolle und einer Gruppe.

Parameter:

groupId

roleId

Ausnahmebehandlung:

SQLException

DatabaseException

SQLException

Definiert in Zeile 734 der Datei DalRole.java.

static void removeAssignedUser (int userId, int roleId) throws DatabaseException, SQLException [static]

Entfernt die Verknuepfung zwischen einer Rolle und einem Benutzer.

Parameter:

userId

roleId

Ausnahmebehandlung:

SQLException

DatabaseException

SQLException

Definiert in Zeile 749 der Datei DalRole.java.

static void updateAccess (long id, String access, boolean metaRight) throws DatabaseException [static]

Überschreibt die Zugriffsrechte auf Daten in der Tablle auth_role_right

Parameter:

id,die Id des Rechts

access,das Recht

Ausnahmebehandlung:

SQLException,DatabaseException

Definiert in Zeile 688 der Datei DalRole.java.

static void updateRole (int roleId, String name, String description) throws DatabaseException, SQLException [static]

Uebernimmt die neuen Werte(Name und Beschreibung) fuer eine Rolle.

Parameter:

roleId

name

description

Ausnahmebehandlung:

DatabaseException

SQLException

Definiert in Zeile 713 der Datei DalRole.java.

1.2.13 DalUser Klassenreferenz

1.2.13.1 Ausführliche Beschreibung

Die Klasse DalUser stellt Methoden zum Bearbeiten und Lesen von Benutzern bereit
Implementiert das Interface IDalUser aus dem Paket auth.api.

Autor:

Christian R.

Definiert in Zeile 26 der Datei DalUser.java.

Öffentliche, statische Methoden

- static void assignGroup (int groupId, int userId) throws DatabaseException
- static void assignRole (int userId, int roleId) throws DatabaseException
- static void createUser (String firstname, String surname, String login, String password, String workstation) throws DatabaseException
- static void deleteUser (int userId) throws SQLException, DatabaseException, NotDeletableException
- static ArrayList< Group > getAssignedGroups (int userId) throws SQLException, DatabaseException
- static ArrayList< Role > getAssignedRoles (int userId) throws SQLException, DatabaseException
- static User getSingleUser (int userId) throws SQLException, DatabaseException
- static User getUser (String userId) throws SQLException, DatabaseException
- static Integer getUserIdByLogin (String login) throws SQLException, DatabaseException
- static ArrayList< User > getUsers () throws SQLException, DatabaseException
- static void removeAssignedGroup (int groupId, int userId) throws SQLException, DatabaseException
- static void removeAssignedRole (int userId, int roleId) throws SQLException, DatabaseException
- static boolean setPassword (String userId, String oldPassword, String newPassword) throws SQLException, DatabaseException
- static void updateUser (int userId, String firstname, String surname, String login, String password, String workstation) throws DatabaseException
- static void updateUserWithoutPassword (int userId, String firstname, String surname, String login, String workstation) throws DatabaseException

1.2.13.2 Dokumentation der Elementfunktionen

static void assignGroup (int groupId, int userId) throws DatabaseException
[static]

Erstellt eine neue Verknuepfung zwischen einem Benutzer und einer Gruppe.

Parameter:

groupId, die Id der Gruppe.

userId, die Id des Benutzers.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 63 der Datei DalUser.java.

static void assignRole (int userId, int roleId) throws DatabaseException [static]

Erstellt eine neue Verknuepfung zwischen einem Benutzer und einer Rolle.

Parameter:

userId, die Id des Benutzers.

roleId, die Id der Rolle.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 83 der Datei DalUser.java.

static void createUser (String firstname, String surname, String login, String password, String workstation) throws DatabaseException [static]

Erstellt in der Datenbank einen Neuen Benutzer.

Parameter:

firstname, der Vorname.

surname, der Nachname.

login, der Login mit dem sich der Benutzer anmelden kann.

password, das Passwort zur Authentifizierung

workstation, der Arbeitsplatz des Benutzers.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 39 der Datei DalUser.java.

static void deleteUser (int userId) throws SQLException, DatabaseException, NotDeletableException [static]

Entfernt einen Benutzer und dessen Verknuepfungen aus dem System.

Parameter:

userId,die Id des Benutzers, der entfernt werden soll.

Ausnahmebehandlung:

SQLException

DatabaseException

NotDeletableException

DatabaseException

NotDeletableException

Definiert in Zeile 244 der Datei DalUser.java.

Benutzt DalUser.getSingleUser().

static ArrayList<Group> getAssignedGroups (int userId) throws SQLException, DatabaseException [static]

Liefert die Gruppen, zu denen ein Benutzer gehoert.

Parameter:

userId,die Id des Nutzers

Rückgabe:

groups, die Liste der Gruppen, denen der Benutzer angehört, mit Id, Name und Beschreibung.

Ausnahmebehandlung:

SQLException,DatabaseException

Definiert in Zeile 103 der Datei DalUser.java.

static ArrayList<Role> getAssignedRoles (int userId) throws SQLException, DatabaseException [static]

Liefert die Rollen, die einem Benutzer zugeordnet sind.

Parameter:

userId,die Id des Benutzers.

Rückgabe:

roles, die Liste der Rollen, denen der Benutzer zugeordnet ist, mit Id, Name und Beschreibung.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 134 der Datei DalUser.java.

static User getSingleUser (int userId) throws SQLException, DatabaseException
[static]

Liefert einen einzelnen Benutzer.

Parameter:

userId, die Id des gewünschten Benutzers.

Rückgabe:

User, das Benutzerobjekt mit Vorname, Nachname, Login-Name, Passwort und Arbeitsplatz.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 163 der Datei DalUser.java.

Wird benutzt von DalUser.deleteUser().

static User getUser (String userId) throws SQLException, DatabaseException
[static]

Liefert einen Benutzer anhand seiner Id.

Parameter:

userId

Rückgabe:

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 188 der Datei DalUser.java.

static Integer getUserIdByLogin (String login) throws SQLException, DatabaseException [static]

Liefert die Id eines Benutzers anhand seines Login-Namens

Parameter:

login,der Login des Benutzers.

Rückgabe:

die Id des Benutzers.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 176 der Datei DalUser.java.

static ArrayList<User> getUsers () throws SQLException, DatabaseException [static]

Liefert alle Benutzer des Systems.

Rückgabe:

Alle Nutzer des Systems mit Id, Vorname, Nachname, Login, Passwort und Arbeitsplatz.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 214 der Datei DalUser.java.

static void removeAssignedGroup (int groupId, int userId) throws SQLException, DatabaseException [static]

Entfernt die Verknuepfung zwischen einer Gruppe und einem Benutzer.

Parameter:

groupId, die Id der Gruppe.

userId, die Id des Benutzers.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 268 der Datei DalUser.java.

static void removeAssignedRole (int userId, int roleId) throws SQLException, DatabaseException [static]

Entfernt die Verknuepfung zwischen einer Rolle und einem Benutzer

Parameter:

userId,die Id des Benutzers.

roleId,die Id der Rolle.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 282 der Datei DalUser.java.

static boolean setPassword (String userId, String oldPassword, String newPassword) throws SQLException, DatabaseException [static]

Setzt ein neues Passwort fuer einen Benutzer.

Parameter:

userId

oldPassword

newPassword

Rückgabe:

false, falls das alte Passwort fehlerhaft ist.

true, falls korrekt.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 299 der Datei DalUser.java.

static void updateUser (int userId, String firstname, String surname, String login, String password, String workstation) throws DatabaseException [static]

Uebernimmt neue Werte fuer einen Benutzer.

Parameter:

userId

firstname

surname

login

password

workstation

Ausnahmebehandlung:

DatabaseException

Definiert in Zeile 361 der Datei DalUser.java.

static void updateUserWithoutPassword (int userId, String firstname, String surname, String login, String workstation) throws DatabaseException [static]

Uebernimmt die neuen Werte fuer einen Benutzer, allerdings ohne ein neues Passwort. Das alte Passwort bleibt also erhalten.

Parameter:

userId

firstname

surname

login

workstation

Ausnahmebehandlung:

DatabaseException

Definiert in Zeile 389 der Datei DalUser.java.

1.2.14 MetamodelAuth Klassenreferenz

1.2.14.1 Ausführliche Beschreibung

Methoden für die Rechte und Rollenverwaltung. Es werden Objekte der Klasse AuthInfo geliefert, die Informationen zu Bestandteilen des Metamodells enthalten.

Autor:

Roland Koppe

Definiert in Zeile 29 der Datei MetamodelAuth.java.

Öffentliche, statische Methoden

- static ArrayList< AuthInfo > getAttributesByAttributeType (long attributeTypeId)
- static ArrayList< AuthInfo > getAttributesByMetamodel (long categoryId)
- static ArrayList< AuthInfo > getAttributesByObject (long objectId)
- static ArrayList< AuthInfo > getAttributesByRelation (long relationId)
- static ArrayList< AuthInfo > getAttributeTypesByObject (long objectId)
- static ArrayList< AuthInfo > getAttributeTypesByRelation (long relationId)

- static ArrayList< AuthInfo > getMetamodels ()
- static AuthInfo getObjectById (long id)
- static ArrayList< AuthInfo > getObjectsByMetamodel (long categoryId)
- static ArrayList< AuthInfo > getObjectsByObjectType (long objectTypeId)
- static ArrayList< AuthInfo > getObjectTypes ()
- static ArrayList< AuthInfo > getObjectTypesByMetamodel (long categoryId)
- static ArrayList< AuthInfo > getRelationsByMetamodel (long categoryId)
- static ArrayList< AuthInfo > getRelationsByRelationType (long relationTypeId)
- static ArrayList< AuthInfo > getRelationTypes ()
- static ArrayList< AuthInfo > getRelationTypesByMetamodel (long categoryId)

1.2.14.2 Dokumentation der Elementfunktionen

static ArrayList<AuthInfo> getAttributesByAttributeType (long attributeTypeId)
[static]

Liefert alle Attribute zu dem angegebenen Typ.

Parameter:

attributeTypeId Attributtyp

Rückgabe:

List<AuthInfo>

Definiert in Zeile 207 der Datei MetamodelAuth.java.

static ArrayList<AuthInfo> getAttributesByMetamodel (long categoryId) [static]

Liefert alle Attribute zurück, die im angegebenen Metamodell verfügbar sind.

Parameter:

categoryId long ID des Metamodells

Rückgabe:

ArrayList<AuthInfo>

Definiert in Zeile 689 der Datei MetamodelAuth.java.

Benutzt DAOFactory.getEAMCategoryDAO(), DAOFactory.getEAMObjectDAO() und DAOFactory.getEAMRelationDAO().

static ArrayList<AuthInfo> getAttributesByObject (long objectId) [static]

Liefert alle Attribute eines Objekts.

Parameter:

objectId Objekt

Rückgabe:

List<AuthInfo>

Definiert in Zeile 241 der Datei MetamodelAuth.java.

static ArrayList<AuthInfo> getAttributesByRelation (long relationId) [static]

Liefert alle Attribute einer Relation.

Parameter:

relationId Relation

Rückgabe:

List<AuthInfo>

Definiert in Zeile 273 der Datei MetamodelAuth.java.

static ArrayList<AuthInfo> getAttributeTypesByObject (long objectId) [static]

Liefert alle Attribute-Typen eines Objekts.

Parameter:

objectId Objekt

Rückgabe:

List<AuthInfo>

Definiert in Zeile 494 der Datei MetamodelAuth.java.

static ArrayList<AuthInfo> getAttributeTypesByRelation (long relationId) [static]

Liefert alle Attribut-Typen einer Relation.

Parameter:

relationId Relation

Rückgabe:

List<AuthInfo>

Definiert in Zeile 537 der Datei MetamodelAuth.java.

static ArrayList<AuthInfo> getMetamodels () [static]

Liefert alle Metamodelle.

Rückgabe:

List<AuthInfo>

Definiert in Zeile 37 der Datei MetamodelAuth.java.

static AuthInfo getObjectById (long id) [static]

Parameter:

id

Rückgabe:

Definiert in Zeile 578 der Datei MetamodelAuth.java.

static ArrayList<AuthInfo> getObjectsByMetamodel (long categoryId) [static]

Liefert alle Objekte des angegebenen Metamodells.

Parameter:

categoryId Metamodel

Rückgabe:

List<AuthInfo>

Definiert in Zeile 68 der Datei MetamodelAuth.java.

static ArrayList<AuthInfo> getObjectsByObjectType (long objectType) [static]

Liefert alle Objekte zu dem angegebenen Typ.

Parameter:

objectType Objekttype

Rückgabe:

List<AuthInfo>

Definiert in Zeile 139 der Datei MetamodelAuth.java.

static ArrayList<AuthInfo> getObjectTypes () [static]

Liefert alle Objekt-Typen aller Metamodelle.

Rückgabe:

List<AuthInfo>

Definiert in Zeile 304 der Datei MetamodelAuth.java.

static ArrayList<AuthInfo> getObjectTypesByMetamodel (long categoryId)
[static]

Liefert alle Objekt-Typen des angegebenen Metamodells.

Parameter:

categoryId Metamodell

Rückgabe:

List<AuthInfo>

Definiert in Zeile 355 der Datei MetamodelAuth.java.

static ArrayList<AuthInfo> getRelationsByMetamodel (long categoryId) [static]

Liefert alle Relationen des angegebenen Metamodells.

Parameter:

categoryId Metamodell

Rückgabe:

List<AuthInfo>

Definiert in Zeile 104 der Datei MetamodelAuth.java.

static ArrayList<AuthInfo> getRelationsByRelationType (long relationTypeId)
[static]

Liefert alle Relationen zu dem angegebenen Typ.

Parameter:

relationTypeId Relationstyp

Rückgabe:

List<AuthInfo>

Definiert in Zeile 173 der Datei MetamodelAuth.java.

static ArrayList<AuthInfo> getRelationTypes () [static]

Liefert alle Relationen-Typen aller Metamodelle.

Rückgabe:

List<AuthInfo>

Definiert in Zeile 397 der Datei MetamodelAuth.java.

static ArrayList<AuthInfo> getRelationTypesByMetamodel (long categoryId)
[static]

Liefert alle Relationen-Typen des angegebenen Metamodells.

Parameter:

categoryId Metamodell

Rückgabe:

List<AuthInfo>

Definiert in Zeile 449 der Datei MetamodelAuth.java.

1.2.15 UserInformation Klassenreferenz

1.2.15.1 Ausführliche Beschreibung

Liefert Informationen zu einem Benutzer.

Autor:

Roland Koppe

Definiert in Zeile 18 der Datei UserInformation.java.

1.2.16 AuthInfo Klassenreferenz

1.2.16.1 Ausführliche Beschreibung

Stellt wesentliche Informationen zu Teilen des Metamodells bereit. Dazu gehören die id, name, description und rights eines Objekts.

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei AuthInfo.java.

Öffentliche Methoden

- AuthInfo (long id, String name, String description, String right, String access)
- String toString ()

1.2.16.2 Beschreibung der Konstruktoren und Destruktoren

AuthInfo (long id, String name, String description, String right, String access)

Konstruktor.

Parameter:

- id** long Id des AuthInfo-Objekts
- name** String Name des AuthInfo-Objekts
- description** String Beschreibung des AuthInfo-Objekts
- right** String Name des Rechts für das Objekt
- String** access Zugriffsrecht auf das Objekt

Definiert in Zeile 49 der Datei AuthInfo.java.

1.2.16.3 Dokumentation der Elementfunktionen**String toString ()**

Gibt den folgenden String zurück: "<id>, <name>, <right>".

Definiert in Zeile 102 der Datei AuthInfo.java.

1.2.17 Group Klassenreferenz**1.2.17.1 Ausführliche Beschreibung****Autor:**

Roland Koppe

Definiert in Zeile 10 der Datei Group.java.

Öffentliche Methoden

- boolean getDeletable ()

1.2.17.2 Dokumentation der Elementfunktionen**boolean getDeletable ()**

Gibt an, ob das Element loeschbar ist oder nicht.

Rückgabe:

true, falls loeschbar false, falls nicht loeschbar

Definiert in Zeile 80 der Datei Group.java.

1.2.18 LoginTag Klassenreferenz

Abgeleitet von javax:servlet:jsp::tagext::TagSupport.

1.2.18.1 Ausführliche Beschreibung

Diese Klasse ist für die Taglib des Sicherheitssystem. Setzt man `loggedIn` auf `yes` dann wird der Body-Bereich des Tags nur angezeigt, wenn ein User eingeloggt ist. Mit Hilfe diesen Tags kann man seine Seite absichern, damit bestimmte Bereiche oder die ganze Seite auch wirklich nur dann angezeigt wird, wenn ein User am System angemeldet ist. Ist `loggedIn` auf `no` dann wird dieser Bereich NUR dann angezeigt, wenn kein Benutzer eingeloggt ist.

Autor:

christian zillmann

Definiert in Zeile 19 der Datei `LoginTag.java`.

1.2.19 RoleTag Klassenreferenz

Abgeleitet von `javax::servlet::jsp::tagext::TagSupport`.

1.2.19.1 Ausführliche Beschreibung

Diese Klasse ist fuer die Taglibrary des Sicherheitssystem. Mit Hilfe dieses Tags werden Bereiche auf einer Seite mit einer Rolle/ID/Recht gekennzeichnet. Diese/s Rolle/ID/Recht wird spaeter von Admin dann auf Rollen gemappt die existieren.

Autor:

christian zillmann

Definiert in Zeile 24 der Datei `RoleTag.java`.

Öffentliche Methoden

- `int doStartTag ()` throws `JspException`
- `void setRequireRole (String requireRole)`

1.2.19.2 Dokumentation der Elementfunktionen

`int doStartTag ()` throws `JspException`

Definiert in Zeile 73 der Datei `RoleTag.java`.

`void setRequireRole (String requireRole)`

Parameter:

`requireRole` the `requireRole` to set

Definiert in Zeile 43 der Datei `RoleTag.java`.

1.2.20 UMUser Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::auth::model::User.

1.2.20.1 Ausführliche Beschreibung

ACHTUNG DIESE KLASSE NICHT EXPORTIEREN! PASSWORT UND CO KÖNNEN GELESEN WERDEN!!!

Autor:

Roland Koppe

Definiert in Zeile 20 der Datei UMUser.java.

1.2.21 UserManager Klassenreferenz

1.2.21.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 30 der Datei UserManager.java.

Öffentliche, statische Methoden

- static boolean changeUserdata (long userId, String password, UMUser user)
- static String encryptPassword (String password)
- static User getSimpleUser (String userId)
- static User getSimpleUser (long userId)
- static UMUser getUser (String userId)
- static UMUser getUser (long userId)
- static UMUser getUser (long userId, String password)
- static HashMap< Long, UMUser > getUsers ()
- static LinkedList< UMUser > getUsersL ()
- static boolean isLoggedIn (long userId, String password)
- static boolean isLoggedIn (long userId)
- static boolean isLoggedIn (String username)
- static LinkedList< Group > loadGroupsByUser (String userId)
- static LinkedList< Group > loadGroupsByUser (long userId)
- static LinkedList< Role > loadRolesByUser (String userId)
- static LinkedList< Role > loadRolesByUser (long userId)
- static UMUser login (String username, String passwordClear)

- static void logout (String userId)
- static void logout (long userId)
- static void reloadGroupRightsForUser (int userId)
- static void reloadInstanceRights (long userId)
- static void reloadMetamodelRights (long userId)
- static void reloadMethodRights (long userId)
- static void reloadRoleRightByGroup (int groupId)
- static void reloadRoleRightsForUser (int userId)
- static void reloadViews (long userId)

1.2.21.2 Dokumentation der Elementfunktionen

static boolean changeUserdata (long userId, String password, UMUser user)
[static]

Ändert die Informationen des angegebenen Benutzers, aber keine Informationen über zugeordnete Gruppen oder Rollen.

Parameter:

userId long ID des Benutzers
password String Passwort
user User bearbeiter Benutzer

Rückgabe:

Definiert in Zeile 272 der Datei UserManager.java.

Benutzt UserManager.encryptPassword(), UMUser.getName(), UMUser.getPasswordClear(), UMUser.getUserId(), UMUser.setName(), UMUser.setPassword() und UMUser.setPasswordClear().

static String encryptPassword (String password) [static]

Berechnet den Hashwert des angegebenen Passworts als MD5.

Autor:

???

Parameter:

password String Passwort

Rückgabe:

String verschlüsseltes Passwort

Definiert in Zeile 845 der Datei UserManager.java.

Wird benutzt von UserManager.changeUserdata().

static User getSimpleUser (String userId) [static]

Liefert das User-Objekt zur angegebenen Benutzer-Id, wenn sich dieses bereits im UserManager befindet.

Parameter:

userId String ID der Benutzers

Rückgabe:

UMUser Objekt

Definiert in Zeile 208 der Datei UserManager.java.

static User getSimpleUser (long userId) [static]

Liefert das User-Objekt zur angegebenen Benutzer-Id, wenn sich dieses bereits im UserManager befindet.

Parameter:

userId long ID der Benutzers

Rückgabe:

UMUser Objekt

Definiert in Zeile 195 der Datei UserManager.java.

static UMUser getUser (String userId) [static]

Liefert das User-Objekt zur angegebenen Benutzer-Id, wenn sich dieses bereits im UserManager befindet.

Parameter:

userId String ID der Benutzers

Rückgabe:

UMUser Objekt

Definiert in Zeile 182 der Datei UserManager.java.

static UMUser getUser (long userId) [static]

Liefert das User-Objekt zur angegebenen Benutzer-Id, wenn sich dieses bereits im UserManager befindet.

Parameter:

userId long ID der Benutzers

Rückgabe:

UMUser Objekt

Definiert in Zeile 169 der Datei UserManager.java.

static UMUser getUser (long userId, String password) [static]

Liefert das User-Objekt zur angegebenen Benutzer-Id, wenn sich dieses bereits im UserManager befindet.

Parameter:

userId long ID des Benutzers

password String Passwort des Benutzers

Rückgabe:

User Objekt zu den angegebenen Daten

Definiert in Zeile 147 der Datei UserManager.java.

Benutzt UMUser.getPassword().

Wird benutzt von UserManager.reloadInstanceRights(),
UserManager.reloadMetamodelRights(), UserManager.reloadMethodRights() und
UserManager.reloadViews().

static HashMap<Long, UMUser> getUsers () [static]

Liefert die User des UserManagers als HashMap zurück.

Rückgabe:

HashMap<Long, User> users

Definiert in Zeile 41 der Datei UserManager.java.

static LinkedList<UMUser> getUsersL () [static]

Liefert die User des UserManagers als LinkedList zurück.

Rückgabe:

LinkedList<User> users

Definiert in Zeile 52 der Datei UserManager.java.

static boolean isLoggedIn (long userId, String password) [static]

Prüft, ob der angegebene Benutzer eingeloggt ist und außerdem das angegebene Passwort dazu passt.

Parameter:

userId long userId

password String Passwort

Rückgabe:

boolean true, wenn Benutzer eingeloggt

Definiert in Zeile 251 der Datei UserManager.java.

Benutzt UMUser.getPassword().

static boolean isLoggedIn (long userId) [static]

Prüft, ob der angegebene Benutzer eingeloggt ist.

Parameter:

userId long userId

Rückgabe:

boolean true, wenn Benutzer eingeloggt

Definiert in Zeile 237 der Datei UserManager.java.

static boolean isLoggedIn (String username) [static]

Prüft, ob der angegebene Benutzer eingeloggt ist.

Parameter:

username String Username

Rückgabe:

boolean true, wenn Benutzer eingeloggt

Definiert in Zeile 220 der Datei UserManager.java.

static LinkedList<Group> loadGroupsByUser (String userId) [static]

Liefert die dem angegebenen Benutzer zugeordneten Gruppen.

Parameter:

userId long Id des Benutzers

Rückgabe:

LinkedList<Group>

Definiert in Zeile 766 der Datei UserManager.java.

Benutzt UserManager.loadGroupsByUser().

static LinkedList<Group> loadGroupsByUser (long userId) [static]

Liefert die dem angegebenen Benutzer zugeordneten Gruppen.

Parameter:

userId long Id des Benutzers

Rückgabe:

LinkedList<Group>

Definiert in Zeile 722 der Datei UserManager.java.

Wird benutzt von UserManager.loadGroupsByUser(), UserManager.login() und UserManager.reloadGroupRightsForUser().

static LinkedList<Role> loadRolesByUser (String userId) [static]

Liefert die dem angegebenen Benutzer zugeordneten Rollen.

Parameter:

userId long Id des Benutzers

Rückgabe:

LinkedList<Role>

Definiert in Zeile 827 der Datei UserManager.java.

Benutzt UserManager.loadRolesByUser().

static LinkedList<Role> loadRolesByUser (long userId) [static]

Liefert die dem angegebenen Benutzer zugeordneten Rollen.

Parameter:

userId long Id des Benutzers

Rückgabe:

LinkedList<Role>

Definiert in Zeile 783 der Datei UserManager.java.

Wird benutzt von UserManager.loadRolesByUser(), UserManager.login() und UserManager.reloadRoleRightsForUser().

static UMUser login (String username, String passwordClear) [static]

Führt den Login für den angegebenen Benutzer durch und liefert das dazugehörige User-Objekt zurück. Des weiteren wird das User-Objekt im UserManager mit dazugehörigen Rechten gespeichert.

Parameter:

username String name

passwordClear String Passwort

Rückgabe:

User

Definiert in Zeile 67 der Datei UserManager.java.

Benutzt UMUser.getId(), UserManager.loadGroupsByUser(), UserManager.loadRolesByUser(), UMUser.setGroups(), UMUser.setInstanceRights(), UMUser.setMetamodelRights(), UMUser.setMethodRights(), UMUser.setRoles() und UMUser.setViews().

static void logout (String userId) [static]

Entfernt das zu der angegebenen Benutzer-ID gehörende User-Objekt aus dem UserManager.

Parameter:

userId String User Id

Definiert in Zeile 128 der Datei UserManager.java.

Benutzt UserManager.logout().

static void logout (long userId) [static]

Entfernt das zu der angegebenen Benutzer-ID gehörende User-Objekt aus dem UserManager.

Parameter:

userId long User Id

Definiert in Zeile 113 der Datei UserManager.java.

Wird benutzt von UserManager.logout().

static void reloadGroupRightsForUser (int userId) [static]

Laedt die Liste der Gruppen und saemtliche Rechte fuer einen Benutzer neu.

Autor:

Christian R.

Parameter:

userId,die Id des geaenderten Nutzers.

Definiert in Zeile 389 der Datei UserManager.java.

Benutzt UserManager.loadGroupsByUser(), UserManager.reloadInstanceRights(), UserManager.reloadMetamodelRights(), UserManager.reloadMethodRights() und UserManager.reloadViews().

static void reloadInstanceRights (long userId) [static]

Lädt die Instanz-Rechte für den angegebenen Benutzer neu.

Parameter:

userId long ID des Benutzers

Definiert in Zeile 296 der Datei UserManager.java.

Benutzt UMUser.addInstanceRights() und UserManager.getUser().

Wird benutzt von UserManager.reloadGroupRightsForUser(), UserManager.reloadRoleRightByGroup() und UserManager.reloadRoleRightsForUser().

static void reloadMetamodelRights (long userId) [static]

Lädt die Metamodell-Rechte für den angegebenen Benutzer neu.

Parameter:

userId long ID des Benutzers

Definiert in Zeile 311 der Datei UserManager.java.

Benutzt UMUser.addMetamodelRights() und UserManager.getUser().

Wird benutzt von UserManager.reloadGroupRightsForUser(), UserManager.reloadRoleRightByGroup() und UserManager.reloadRoleRightsForUser().

static void reloadMethodRights (long userId) [static]

Lädt die Methoden-Rechte für den angegebenen Benutzer neu.

Parameter:

userId long ID des Benutzers

Definiert in Zeile 326 der Datei UserManager.java.

Benutzt UMUser.addMethodRights(), UserManager.getUser() und UMUser.getUserId().

Wird benutzt von UserManager.reloadGroupRightsForUser(), UserManager.reloadRoleRightByGroup() und UserManager.reloadRoleRightsForUser().

static void reloadRoleRightByGroup (int groupId) [static]

Lädt die Rechte aller Benutzer neu, die der uebergebenen GruppenId angehören.

Autor:

Christian R.

Parameter:

groupId,die Id der geaenderten Gruppe

deleted,true falls die Gruppe geloescht wird sonst false.

Definiert in Zeile 369 der Datei UserManager.java.

Benutzt UserManager.reloadInstanceRights(), UserManager.reloadMetamodelRights(), UserManager.reloadMethodRights() und UserManager.reloadViews().

static void reloadRoleRightsForUser (int userId) [static]

Laedt die Liste der Rollen eines Nutzers neu und dann auch die Rechte.

Autor:

Christian R.

Parameter:

userId,die Id des geaenderten Benutzers

Definiert in Zeile 408 der Datei UserManager.java.

Benutzt UserManager.loadRolesByUser(), UserManager.reloadInstanceRights(), UserManager.reloadMetamodelRights(), UserManager.reloadMethodRights() und UserManager.reloadViews().

static void reloadViews (long userId) [static]

Lädt die Views für den angegebenen Benutzer neu.

Parameter:

userId long ID des Benutzers

Definiert in Zeile 345 der Datei UserManager.java.

Benutzt UMUser.addViewes(), UserManager.getUser() und UMUser.getUserId().

Wird benutzt von UserManager.reloadGroupRightsForUser(), UserManager.reloadRoleRightByGroup() und UserManager.reloadRoleRightsForUser().

1.3 beans

1.3.1 UserInfoBean Klassenreferenz

1.3.1.1 Ausführliche Beschreibung

Diese Bean gibt Informationen über einen User raus, damit die dann auf der Startseite angezeigt werden können.

Autor:

christian zillmann

Definiert in Zeile 21 der Datei UserInfoBean.java.

1.3.2 UserManagerBean Klassenreferenz

1.3.2.1 Ausführliche Beschreibung

Diese Bean ermöglicht das Abmelden von Benutzern und das Neuladen von Rechten für Instanzen des Metamodells durch den Administrator.

Autor:

Roland Koppe

Definiert in Zeile 24 der Datei UserManagerBean.java.

1.3.3 BundleInformationBean Klassenreferenz

1.3.3.1 Ausführliche Beschreibung

Diese Bean liefert Informationen über die aktuell gestarteten Bundles, die mit dem Prefix "EAM" beginnen.

Autor:

Roland Koppe

Definiert in Zeile 19 der Datei BundleInformationBean.java.

1.3.4 ContainerBean Klassenreferenz

1.3.4.1 Ausführliche Beschreibung

Autor:

Christian Zillmann Diese Klasse dient dazu, den Inhalt des Containers ueber JSF auszugeben.

Definiert in Zeile 17 der Datei ContainerBean.java.

1.3.5 ImportExportBean Klassenreferenz

1.3.5.1 Ausführliche Beschreibung

Diese Bean bietet die Benutzungsschnittstelle zum Import und Export des Cores.

Autor:

Roland Koppe

Definiert in Zeile 39 der Datei ImportExportBean.java.

Öffentliche Methoden

- void exportMetamodel ()
- ImportExportBean ()
- boolean importMetamodel ()

1.3.5.2 Beschreibung der Konstruktoren und Destruktoren

ImportExportBean ()

Standard-Konstruktor.

Definiert in Zeile 55 der Datei ImportExportBean.java.

1.3.5.3 Dokumentation der Elementfunktionen

void exportMetamodel ()

Exportiert das ausgewählte Metamodell

Definiert in Zeile 78 der Datei ImportExportBean.java.

Benutzt Export.addExportElement() und Export.export().

boolean importMetamodel ()

Importiert im Request angegebene "metamodelFile" und "instanceFile".

Definiert in Zeile 220 der Datei ImportExportBean.java.

Benutzt Import.importInstancesAsNew() und Import.importMetamodelAsNew().

1.3.6 LoggingBean Klassenreferenz

1.3.6.1 Ausführliche Beschreibung

Diese Bean zeigt die bisherigen Logs des EAM-Tools an.

Autor:

Roland Koppe

Definiert in Zeile 21 der Datei LoggingBean.java.

1.3.7 ResourceServlet Klassenreferenz

Abgeleitet von javax::servlet::http::HttpServlet.

1.3.7.1 Ausführliche Beschreibung

Dieses Servlet lädt die über den Parameter "filename" angegebene Ressource als mit dem Parameter "contentType" angegebenen Inhalt und liefert das Ergebnis im ServletOutputStream.

Autor:

Roland Koppe

Definiert in Zeile 25 der Datei ResourceServlet.java.

1.3.8 Resources Klassenreferenz

1.3.8.1 Ausführliche Beschreibung

Diese Klasse verwaltet freie Ressourcen, die auf dem Server zur Verfügung gestellt werden. Das ResourceServlet kann auf diese Ressourcen verweisen und somit Benutzern das herunterladen von Dateien ermöglichen.

Autor:

Roland Koppe

Definiert in Zeile 15 der Datei Resources.java.

Öffentliche, statische Methoden

- static void addResource (String resource)
- static void addResources (String resources)
- static HashSet< String > getResources ()
- static boolean isResource (String resource)
- static void removeResouce (String resource)

1.3.8.2 Dokumentation der Elementfunktionen

static void addResource (String resource) [static]

Gibt die angegebene Ressource frei. Es sind auch Angaben in der Form `/home/xya/*` möglich, damit werden alle Ressourcen im Verzeichnis `xya` freigegeben.

Parameter:

resource String Datei oder Pfad

Definiert in Zeile 64 der Datei `Resources.java`.

static void addResources (String resources) [static]

Gibt die angegebenen Ressourcen frei. Dabei werden einzelne Ressourcen durch Semikolon (;) getrennt.

Parameter:

resources String

Definiert in Zeile 76 der Datei `Resources.java`.

static HashSet<String> getResources () [static]

Liefert alle Ressourcen zurück.

Rückgabe:

HashSet<String>

Definiert in Zeile 103 der Datei `Resources.java`.

static boolean isResource (String resource) [static]

Liefert zurück, ob die angegebene Ressource außerhalb des Servers sichtbar ist.

Parameter:

resource

Rückgabe:

Definiert in Zeile 28 der Datei `Resources.java`.

static void removeResource (String resource) [static]

Entfernt die angegebene Ressource.

Parameter:

resource String Datei oder Pfad

Definiert in Zeile 92 der Datei `Resources.java`.

1.3.9 MultipartWrapper Klassenreferenz

1.3.9.1 Ausführliche Beschreibung

Diese Klasse ermöglicht das Auslesen von im HttpServletRequest angehängenen Daten z.B. bei POST und type="multipart". Im Speziellen können damit Dateien auf den Server geladen werden.

Autor:

Roland Koppe

Definiert in Zeile 20 der Datei MultipartWrapper.java.

Öffentliche Methoden

- String getCharset ()
- String getContentType ()
- HashMap< String, Part > getParts ()
- MultipartWrapper (HttpServletRequest request) throws IOException

1.3.9.2 Beschreibung der Konstruktoren und Destruktoren

MultipartWrapper (HttpServletRequest request) throws IOException

Konstruktor.

Parameter:

request HttpServletRequest

Ausnahmebehandlung:

IOException, wenn InputStream nicht gelesen werden kann

Definiert in Zeile 37 der Datei MultipartWrapper.java.

1.3.9.3 Dokumentation der Elementfunktionen

String getCharset ()

Liefert den Charset des Request.

Rückgabe:

String Charset

Definiert in Zeile 83 der Datei MultipartWrapper.java.

String getContentType ()

Liefert den ContentType des Request.

Rückgabe:

String ContentType

Definiert in Zeile 94 der Datei MultipartWrapper.java.

HashMap<String, Part> getParts ()

Liefert eine HashMap mit allen im Request angegebenen Feldnamen und den dazugehörigen Inhalt.

Rückgabe:

HashMap<String, Part> Feldname, Inhalt

Definiert in Zeile 249 der Datei MultipartWrapper.java.

1.3.10 Part Klassenreferenz**1.3.10.1 Ausführliche Beschreibung**

Diese Klasse definiert einen Daten-Abschnitt im Request. Dazu gehören etwa Feldname und Inhalt des Feldes.

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei Part.java.

Öffentliche Methoden

- String getCharset ()
- String getContentDisposition ()
- String getContentType ()
- String getData ()
- String getFilename ()
- String getName ()
- boolean isFile ()
- void setCharset (String charset)
- void setContentDisposition (String contentDisposition)
- void setContentType (String contentType)
- void setData (String data)
- void setFilename (String filename)

- void setName (String name)

1.3.10.2 Dokumentation der Elementfunktionen

String getCharset ()

Liefert den Charset zurück.

Rückgabe:

String Charset

Definiert in Zeile 136 der Datei Part.java.

String getContentDisposition ()

Liefert die ContentDisposition zurück.

Rückgabe:

String ContentDisposition

Definiert in Zeile 28 der Datei Part.java.

String getContentType ()

Liefert den ContentType zurück.

Rückgabe:

String ContentType

Definiert in Zeile 48 der Datei Part.java.

String getData ()

Liefert die Daten zurück.

Rückgabe:

String Daten

Definiert in Zeile 116 der Datei Part.java.

String getFilename ()

Liefert den Dateinamen, wenn vorhanden.

Rückgabe:

String, "" wenn nicht vorhanden

Definiert in Zeile 87 der Datei Part.java.

String getName ()

Liefert den Namen des Feldes.

Rückgabe:

String Name

Definiert in Zeile 68 der Datei Part.java.

boolean isFile ()

Liefert true, wenn ein Dateiname angegeben wurde.

Rückgabe:

true, wenn Dateiname angegeben

Definiert in Zeile 105 der Datei Part.java.

void setCharset (String charset)

Setzt den Charset.

Parameter:

charset String

Definiert in Zeile 145 der Datei Part.java.

void setContentDisposition (String contentDisposition)

Setzt die ContentDisposition.

Parameter:

contentDisposition String

Definiert in Zeile 37 der Datei Part.java.

void setContentType (String contentType)

Setzt den ContentType.

Parameter:

contentType String

Definiert in Zeile 57 der Datei Part.java.

void setData (String data)

Setzt die Daten des Abschnitts.

Parameter:

data String

Definiert in Zeile 125 der Datei Part.java.

void setFilename (String filename)

Setzt den Dateinamen.

Parameter:

filename String

Definiert in Zeile 96 der Datei Part.java.

void setName (String name)

Setzt den Namen des Feldes.

Parameter:

name String

Definiert in Zeile 76 der Datei Part.java.

1.3.11 UploadServlet Klassenreferenz

Abgeleitet von javax::servlet::http::HttpServlet.

1.3.11.1 Ausführliche Beschreibung

Dieses Servlet ermöglicht das Laden von Dateien auf den Server.

Dabei werden Daten (also auch Dateien) mit einem HTML-Form-Tag mit dem enctype "multipart/form-data" per POST gesendet. Das Servlet speichert die enthaltenen Dateien temporär auf dem Server und schreibt den absoluten Pfad in die Session (upload.<Name des Input-Feldes>). Danach wird wieder auf die Ausgangsseite verwiesen mit dem Parameter &upload=success, wenn das Hochladen erfolgreich war. Sonst ohne weitere Parameter.

Autor:

Roland Koppe

Definiert in Zeile 32 der Datei UploadServlet.java.

1.4 configuration

1.4.1 IBundleConfiguration Schnittstellenreferenz

Basisklasse für Configuration.

1.4.1.1 Ausführliche Beschreibung

Interface, das von einem bundle implementiert werden muss, wenn es eigene Rollen und/oder Objekte und Relationen mitbringt. Bei den boolean-Methoden reicht es aus, true oder false zurückzugeben, falls das Bundle Objekte, Relationen oder Rollen mitbringt.

getObjectsJdomDocument() und getRolesJdomDocument() lesen ihre jeweilige XML-Datei ein und geben dieses als JDOM-Document zurück.

Autor:

Jens Henkel

Definiert in Zeile 19 der Datei IBundleConfiguration.java.

1.4.2 JdomXMLFileReader Klassenreferenz

1.4.2.1 Ausführliche Beschreibung

Autor:

Jens Henkel

Definiert in Zeile 18 der Datei JdomXMLFileReader.java.

Öffentliche, statische Methoden

- static Document getJDOMDocument (InputStream is) throws JDOMException, IOException
- static Document getJDOMDocument (String filepath) throws JDOMException, IOException

1.4.2.2 Dokumentation der Elementfunktionen

static Document getJDOMDocument (InputStream is) throws JDOMException, IOException [static]

Verarbeitet einen InputStream, der eine XML-Datei eingelesen hat und erstellt daraus ein JDOM-Document.

Parameter:

is Der InputStream der eingelesenen XML-Datei.

Rückgabe:

Das JDOM-Document.

Ausnahmebehandlung:

JDOMException

IOException

Definiert in Zeile 47 der Datei JdomXMLFileReader.java.

static Document getJDOMDocument (String filepath) throws JDOMException, IOException [static]

Liest eine XML-Datei ein und erstellt daraus ein JDOM-Document.

Parameter:

filepath Der Pfad zur XML-Datei, die eingelesen werden soll.

Rückgabe:

Das eingelesene JDOM-Document der XML-datei.

Ausnahmebehandlung:

JDOMException

IOException

Definiert in Zeile 30 der Datei JdomXMLFileReader.java.

1.4.3 EAMObjectAttributeMatchingStore Klassenreferenz

1.4.3.1 Ausführliche Beschreibung

Diese Klasse dient der Informationsspeicherung für das Objek-Attribut-Mapping während der Konfiguration der Bundles, um nach einem Mapping nicht ständig Abfragen an die Datenbank starten zu müssen. Nach jedem Mapping wird ein neues Objekt dieser Klasse erzeugt und in eine Liste gelegt. Nach dem Löschen eines Mappings wird dieser aus der vorhandenen Liste gelöscht.

Autor:

Jens Henkel

Definiert in Zeile 14 der Datei EAMObjectAttributeMatchingStore.java.

1.4.4 EAMObjectMatchingStore Klassenreferenz

1.4.4.1 Ausführliche Beschreibung

Diese Klasse dient der Informationsspeicherung für das Objek-Mapping während der Konfiguration der Bundles, um nach einem Mapping nicht ständig Abfragen an die Datenbank starten zu müssen. Nach jedem Mapping wird ein neues Objekt dieser Klasse erzeugt und in eine Liste gelegt. Nach dem Löschen eines Mappings wird dieser aus der vorhandenen Liste gelöscht.

Autor:

Jens Henkel

Definiert in Zeile 13 der Datei EAMObjectMatchingStore.java.

1.4.5 EAMRelationAttributeMatchingStore Klassenreferenz

1.4.5.1 Ausführliche Beschreibung

Diese Klasse dient der Informationsspeicherung für das Relationen-Attribut-Mapping während der Konfiguration der Bundles, um nach einem Mapping nicht ständig Abfragen an die Datenbank starten zu müssen. Nach jedem Mapping wird ein neues Objekt dieser Klasse erzeugt und in eine Liste gelegt. Nach dem Löschen eines Mappings wird dieser aus der vorhandenen Liste gelöscht.

Autor:

Jens Henkel

Definiert in Zeile 13 der Datei EAMRelationAttributeMatchingStore.java.

1.4.6 EAMRelationMatchingStore Klassenreferenz

1.4.6.1 Ausführliche Beschreibung

Diese Klasse dient der Informationsspeicherung für das Relationen-Mapping während der Konfiguration der Bundles, um nach einem Mapping nicht ständig Abfragen an die Datenbank starten zu müssen. Nach jedem Mapping wird ein neues Objekt dieser Klasse erzeugt und in eine Liste gelegt. Nach dem Löschen eines Mappings wird dieser aus der vorhandenen Liste gelöscht.

Autor:

Jens Henkel

Definiert in Zeile 15 der Datei EAMRelationMatchingStore.java.

1.4.7 EAMViewRoleMatchingStore Klassenreferenz

1.4.7.1 Ausführliche Beschreibung

Diese Klasse dient der Informationsspeicherung für das Rollen-Sichten-Mapping während der Konfiguration der Bundles, um nach einem Mapping nicht ständig Abfragen an die Datenbank starten zu müssen. Nach jedem Mapping wird ein neues Objekt dieser Klasse erzeugt und in eine Liste gelegt. Nach dem Löschen eines Mappings wird dieser aus der vorhandenen Liste gelöscht.

Autor:

Jens Henkel

Definiert in Zeile 18 der Datei EAMViewRoleMatchingStore.java.

1.4.8 ModuleToCoreObjectWrapper Klassenreferenz

1.4.8.1 Ausführliche Beschreibung

Statische Klasse, die den Zugriff auf verknüpfte EAM-Objekte und -Relationen ermöglichen soll. Ist vermutlich unvollständig, da bisher auf Grund mangelnder Möglichkeiten zum Testen kein Einsatz möglich ist.

Autor:

Jens Henkel

Definiert in Zeile 27 der Datei ModuleToCoreObjectWrapper.java.

Öffentliche, statische Methoden

- static boolean deleteDataObject (DataObject dataObject)
- static boolean deleteDataObjects (ArrayList< DataObject > dataObjects)
- static LinkedList< Long > findAllMappedEAMObjectIdsByBundleId (long bundleEquinoxId)
- static LinkedList< Long > findAllMappedEAMRelationIdsByBundleId (long bundleEquinoxId)
- static EAMAttribute findObjectAttributeByBundleRelationAndAttributeId (String bundleObjectId, String attributeName)
- static EAMObject findOneCoreObjectByByBundleObjectId (String bundleObjectId)
- static EAMRelation findOneCoreRelationByBundleRelationId (String bundleRelationId)
- static EAMAttribute findRelationAttributeByBundleRelationAndAttributeId (String bundleRelationId, String attributeName)
- static ArrayList< DataObject > loadDataObjects (long coreEamObjectId)

- static boolean saveDataObject (DataObject dataObject)
- static boolean saveDataObjects (ArrayList< DataObject > dataObjects)

1.4.8.2 Dokumentation der Elementfunktionen

static boolean deleteDataObject (DataObject dataObject) [static]

Löscht eine Instanz eines EAM-Objektes aus der Datenbank

Parameter:

dataObjects Die Instanz eines EAM-Objektes, das gelöscht werden soll.

Rückgabe:

true, wenn das Objekt gelöscht werden konnte, sonst false;

Definiert in Zeile 89 der Datei ModuleToCoreObjectWrapper.java.

static boolean deleteDataObjects (ArrayList< DataObject > dataObjects) [static]

Löscht alle Instanzen, die zu den jeweiligen EAM-Objekten in der ArrayList gehören, aus der Datenbank.

Parameter:

dataObjects Eine Liste von Instanzen von EAM-Objektes, die gelöscht werden sollen.

Rückgabe:

true, wenn alle Objekte gelöscht werden konnten. Tritt ein Fehler auf, wird der Vorgang abgebrochen und false zurückgegeben.

Definiert in Zeile 150 der Datei ModuleToCoreObjectWrapper.java.

static LinkedList<Long> findAllMappedEAMObjectIdsByBundleId (long bundleEquinoxId) [static]

Findet alle gemappten EAM-Objekte des Kerns und eines Bundles mit Hilfe der übergebenen Equinox-ID.

Parameter:

bundleId Die durch Equinox vergebene BundleId, um alle EAM-Objekte zu dieser Bundler-ID zu bekommen.

Rückgabe:

Eine Liste aller verknüpfter Objekte für dieses Bundle oder null im Fehlerfall.

Definiert in Zeile 184 der Datei ModuleToCoreObjectWrapper.java.

static LinkedList<Long> findAllMappedEAMRelationIdsByBundleId (long bundleEquinoxId) [static]

Findet alle gemappten EAM-Relation des Kerns und eines Bundles mit Hilfe der übergebenen Equinox-ID.

Parameter:

bundleId Die durch Equinox vergebene BundleId, um alle EAM-Relationen zu dieser Bundler-ID zu bekommen.

Rückgabe:

Eine Liste aller verknüpfter Relationen für dieses Bundle oder null im Fehlerfall.

Definiert in Zeile 199 der Datei ModuleToCoreObjectWrapper.java.

static EAMAttribute findObjectAttributeByBundleRelationAndAttributeId (String bundleObjectId, String attributeName) [static]

Ermittelt ein EAMAttribute einer verknüpften Kern-Relation und einer Bundle-Relation und liefert dieses zurück.

Parameter:

bundleObjectId Die zusammengesetzte und eindeutige ID eines Bundle-Objektes.

attributeName Der Name des Attributes, das zurückgegeben werden soll.

Rückgabe:

Das gefundene EAMAttribute oder null.

Definiert in Zeile 285 der Datei ModuleToCoreObjectWrapper.java.

Benutzt ModuleToCoreObjectWrapper.findOneCoreObjectByBundleObjectId().

static EAMObject findOneCoreObjectByBundleObjectId (String bundleObjectId) [static]

Ermittelt zu der übergebenen ID eines Bundle-Objektes ein EAMObjekt aus dem Kern und liefert dieses zurück.

Parameter:

bundleObjectId Die eindeutige ID eines Bundle-Objektes. Zusammengesetzt aus der Equinox-Id + _ + Objektname + _ + Objekt-ID.

Rückgabe:

Das gefundene EAMObjekt oder null.

Definiert in Zeile 213 der Datei ModuleToCoreObjectWrapper.java.

Wird benutzt von

ModuleToCoreObjectWrapper.findObjectAttributeByBundleRelationAndAttributeId().

static EAMRelation findOneCoreRelationByBundleRelationId (String bundleRelationId) [static]

Ermittelt zu der übergebenen ID einer Bundle-Relation die entsprechende EAMRelation aus dem Kern und liefert diese zurück.

Parameter:

bundleRelationId Die eindeutige ID einer Bundle-Relation. Zusammengesetzt aus der Equinox-Id + _ + Relationsname + _ + Relations-ID.

Rückgabe:

Die gefundenen EAMRelation oder null.

Definiert in Zeile 237 der Datei ModuleToCoreObjectWrapper.java.

Wird benutzt von

ModuleToCoreObjectWrapper.findRelationAttributeByBundleRelationAndAttributeId().

static EAMAttribute findRelationAttributeByBundleRelationAndAttributeId (String bundleRelationId, String attributeName) [static]

Ermittelt ein EAMAttribute einer verknüpften Kern-Relation und einer Bundle-Relation und liefert dieses zurück.

Parameter:

bundleRelationId Die zusammengesetzte und eindeutige ID einer Bundle-Relation.

attributeName Der Name des Attributes, das zurückgegeben werden soll.

Rückgabe:

Das gefundene EAMAttribute oder null.

Definiert in Zeile 262 der Datei ModuleToCoreObjectWrapper.java.

Benutzt ModuleToCoreObjectWrapper.findOneCoreRelationByBundleRelationId().

static ArrayList<DataObject> loadDataObjects (long coreEamObjectId) [static]

Lädt alle Instanzen eines EAM-Objektes abhängig von der ID des EAM-Objektes.

Parameter:

coreEamObjectId Die ID des EAM-Objektes.

Rückgabe:

Eine Liste aller Instanzen, die zu dem jeweiligen EAM-Objekt gehören.

Definiert in Zeile 37 der Datei ModuleToCoreObjectWrapper.java.

static boolean saveDataObject (DataObject dataObject) [static]

Speichert eine Instanz eines EAM-Objektes in der Datenbank

Parameter:

dataObject Die Instanz eines EAM-Objektes, das gespeichert werden soll.

Rückgabe:

true, wenn das Objekt gespeichert werden konnte, sonst false;

Definiert in Zeile 67 der Datei ModuleToCoreObjectWrapper.java.

static boolean saveDataObjects (ArrayList< DataObject > dataObjects) [static]

Speichert alle Instanzen von EAM-Objekten in der Datenbank. Dabei kann es sich um neue Objekte oder nur um aktualisierte handeln.

Parameter:

dataObjects Eine Liste von Instanzen von EAM-Objekten, die gespeichert werden sollen. , wenn alle Objekte gelöscht werden konnten. Tritt ein Fehler auf, wird der Vorgang abgebrochen und false zurückgegeben.

Definiert in Zeile 116 der Datei ModuleToCoreObjectWrapper.java.

1.5 container

1.5.1 ContainerHandlerMemory Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::container::api::IContainerHandler.

1.5.1.1 Ausführliche Beschreibung

Autor:

christian zillmann Ein weiterer ContainerHandler statt in der Datenbank wird der Inhalt im Hauptspeicher gehalten.

Definiert in Zeile 17 der Datei ContainerHandlerMemory.java.

Öffentliche Methoden

- Boolean removeObject (String user, String id)

1.5.1.2 Dokumentation der Elementfunktionen

Boolean removeObject (String user, String id)

Für Module verwendbare remove Methode

Parameter:

user

obj

Rückgabe:

Definiert in Zeile 144 der Datei ContainerHandlerMemory.java.

1.5.2 ContainerHandlerSql Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::container::api::IContainerHandler.

1.5.2.1 Ausführliche Beschreibung

Autor:

Christian Zillmann Diese Klasse ist eine Schnittstelle für einen Container, der seine Objekte persistent in der Datenbank speichert.

Definiert in Zeile 31 der Datei ContainerHandlerSql.java.

Öffentliche Methoden

- Boolean addObject (Prefix praefix, String source, String description, Object obj, String user)
- Boolean clear (String user)
- ArrayList< ContainerObject > getObjects (String user)
- Boolean removeObject (String id)

1.5.2.2 Dokumentation der Elementfunktionen

Boolean addObject (Prefix praefix, String source, String description, Object obj, String user)

Diese Methode fuegt ein Object in den Container ein. Dabei wird die Quelle uebergeben, also wer dieses Object in den Container gestellt hat. Dann kann eine Beschreibung uebergeben werden, um was es sich bei dem Objekt handelt. Zu guter letzt wird dann noch das Objekt uebergeben, was in den Container gepackt werden soll. Wichtig ist hier, dass man darauf achtet, dass dieses Objekt auch serialisierbar ist, denn sonst koennte man es ja nicht persistent speichern.

Definiert in Zeile 43 der Datei ContainerHandlerSql.java.

Boolean clear (String user)

Diese Methode leert den Container des aktuell eingeloggten Users.

Definiert in Zeile 208 der Datei ContainerHandlerSql.java.

ArrayList<ContainerObject> getObjects (String user)

Diese Methode liefert alle ContainerObjekte des aktuell eingeloggten Users zurueck.

Definiert in Zeile 150 der Datei ContainerHandlerSql.java.

Boolean removeObject (String id)

Diese Methode loescht ein Objekt aus dem Container. Uebergeben werden muss die id des Objektes.

Definiert in Zeile 183 der Datei ContainerHandlerSql.java.

1.5.3 ObjectWrapper Klassenreferenz

Abgeleitet von java.io.Serializable.

1.5.3.1 Ausführliche Beschreibung

Die Klasse wrappt ein Object, um es dann in die Datenbank zu schreiben.

Autor:

Christian Zillmann

Definiert in Zeile 10 der Datei ObjectWrapper.java.

1.5.4 ContainerObject Klassenreferenz

Abgeleitet von java.io.Serializable.

1.5.4.1 Ausführliche Beschreibung

Autor:

Roland Koppe, Christian Zillmann

Definiert in Zeile 15 der Datei ContainerObject.java.

Öffentliche Methoden

- ContainerObject (String id, String user, String prefix, String description, String source, Timestamp date, String type, Object object)

1.5.4.2 Beschreibung der Konstruktoren und Destruktoren

ContainerObject (String id, String user, String prefix, String description, String source, Timestamp date, String type, Object object)

Parameter:

id

source

type MIME Type <http://de.selfhtml.org/diverses/mimetypen.htm>

object

Definiert in Zeile 49 der Datei ContainerObject.java.

1.6 database

1.6.1 DatabaseConnectionException Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::database::exceptions::DatabaseException`.

1.6.1.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei DatabaseConnectionException.java.

1.6.2 DatabaseException Klassenreferenz

Basisklasse für DatabaseConnectionException, DatabaseResultSetException, DatabaseSQLException und DatabaseStatementException.

1.6.2.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 9 der Datei DatabaseException.java.

1.6.3 DatabaseResultSetException Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::database::exceptions::DatabaseException`.

1.6.3.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 9 der Datei DatabaseResultSetException.java.

1.6.4 DatabaseSQLException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::database::exceptions::DatabaseException.

1.6.4.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 11 der Datei DatabaseSQLException.java.

1.6.5 DatabaseStatementException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::database::exceptions::DatabaseException.

1.6.5.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 11 der Datei DatabaseStatementException.java.

1.6.6 Connection Klassenreferenz

1.6.6.1 Ausführliche Beschreibung

Stellt die Verbindung zur Datenbank bereit. Die Standard-Einstellungen für die Verbindung sollen mit Connection.setup() gesetzt werden. Nach der erfolgreichen Verbindung kann mit Connection.getConnection() die Verbindung verwendet werden.

Autor:

Roland Koppe

Definiert in Zeile 24 der Datei Connection.java.

Öffentliche, statische Methoden

- static synchronized void close ()

- static synchronized boolean connect () throws SQLException, DatabaseConnectionException
- static java.sql.Connection getConnection () throws DatabaseException
- static String getDatabase ()
- static Driver getDbDriver ()
- static boolean isClosed ()
- static synchronized void setDatabase (String database)
- static void setDbDriverClass (String driver)
- static void setup (String driver, String dbSchema, String dbServer, String dbUser, String dbPassword)
- static void setup (String dbSchema, String dbServer, String dbUser, String dbPassword)

1.6.6.2 Dokumentation der Elementfunktionen

static synchronized void close () [static]

Schließt die Datenbankverbindung.

Definiert in Zeile 196 der Datei Connection.java.

Benutzt Connection.close().

Wird benutzt von Connection.close().

static synchronized boolean connect () throws SQLException, DatabaseConnectionException [static]

Stellt die Verbindung zur Datenbank her.

Ausnahmebehandlung:

SQLException, wenn SQL Fehler

Rückgabe:

true, wenn erfolgreich, sonst false

Ausnahmebehandlung:

DatabaseConnectionException

Definiert in Zeile 123 der Datei Connection.java.

Benutzt Connection.getConnection() und Connection.setDatabase().

Wird benutzt von Connection.getConnection().

static java.sql.Connection getConnection () throws DatabaseException [static]

Liefert die Datenbankverbindung.

Ausnahmebehandlung:

DatabaseException

Rückgabe:

java.sql.Connection, sonst null

Definiert in Zeile 220 der Datei Connection.java.

Benutzt Connection.connect().

Wird benutzt von Connection.connect().

static String getDatabase () [static]

Liefert den Namen des Katalogs der Verbindung, wenn dieser zuvor gesetzt worden ist.

Rückgabe:

String

Definiert in Zeile 294 der Datei Connection.java.

static Driver getDbDriver () [static]

Liefert den Treiber für die Datenbankverbindung.

Rückgabe:

Driver

Definiert in Zeile 260 der Datei Connection.java.

static boolean isClosed () [static]

Gibt zurück, ob die Datenbankverbindung geschlossen ist.

Rückgabe:

true, Verbindung geschlossen

Definiert in Zeile 176 der Datei Connection.java.

Benutzt Connection.isClosed().

Wird benutzt von Connection.isClosed().

static synchronized void setDatabase (String database) [static]

Setzt den Katalog für die Datenbankverbindung.

Parameter:

database String

Definiert in Zeile 271 der Datei Connection.java.

Wird benutzt von Connection.connect().

static void setDbDriverClass (String driver) [static]

Setzt den Treiber für die Datenbankverbindung als qualifizierten Namen der Klasse. Beispielsweise kann die Klasse com.mysql.jdbc.Driver benutzt werden.

Parameter:

driver String com.mysql.jdbc.Driver

Definiert in Zeile 57 der Datei Connection.java.

Wird benutzt von Connection.setup().

static void setup (String driver, String dbSchema, String dbServer, String dbUser, String dbPassword) [static]

Setzt die Standardeinstellungen für die Verbindung.

Parameter:

driver String com.mysql.jdbc.Driver

dbSchema String Schema jdbc:mysql:

dbServer String Server 134.106.56.251:3307

dbUser String Benutzername

dbPassword String Passwort

Definiert in Zeile 89 der Datei Connection.java.

Benutzt Connection.setDbDriverClass() und Connection.setup().

static void setup (String dbSchema, String dbServer, String dbUser, String dbPassword) [static]

Setzt die Standardeinstellungen für die Verbindung.

Parameter:

dbSchema String Schema jdbc:mysql:

dbServer String Server 134.106.56.251:3307

dbUser String Benutzername

dbPassword String Passwort

Definiert in Zeile 71 der Datei Connection.java.

Wird benutzt von Connection.setup().

1.6.7 ConnectionHolder Klassenreferenz

1.6.7.1 Ausführliche Beschreibung

Der ConnectionHolder sendet in angegebenen Abständen eine Anfrage an die Datenbank, um die Verbindung aufrecht zu erhalten.

Autor:

Roland Koppe

Definiert in Zeile 16 der Datei ConnectionHolder.java.

Öffentliche Methoden

- void run ()

Öffentliche, statische Methoden

- static long getWaittime ()
- static void setWaittime (String waittime)

1.6.7.2 Dokumentation der Elementfunktionen

static long getWaittime () [static]

Liefert die Wartezeit bis zum Senden der nächsten Anfrage in Sekunden.

Rückgabe:

long Wartezeit in Sekunden

Definiert in Zeile 50 der Datei ConnectionHolder.java.

void run ()

Startet den ConnectionHolder.

Definiert in Zeile 26 der Datei ConnectionHolder.java.

static void setWaittime (String waittime) [static]

Setzt die Wartezeit in Sekunden bis zur nächsten Anfrage.

Parameter:

waittime String Wartezeit in Sekunden

Definiert in Zeile 62 der Datei ConnectionHolder.java.

1.6.8 Database Klassenreferenz

1.6.8.1 Ausführliche Beschreibung

Diese Klasse dient dem Ausführen von Anfragen an die Datenbank.

<http://dev.mysql.com/doc/refman/5.0/en/connector-j-usagenotes-troubleshooting.html#qandaitem-26-4-5-3-4>

<http://dev.mysql.com/doc/refman/5.1/en/error-messages-server.html>

Autor:

Roland Koppe

Definiert in Zeile 29 der Datei Database.java.

Öffentliche, statische Methoden

- static boolean createDatabase (String database) throws DatabaseException
- static boolean deleteDatabase (String database) throws DatabaseException
- static synchronized ArrayList< Integer > execute (String sql) throws DatabaseException
- static boolean existsTable (String table) throws DatabaseException
- static List< String > getDatabaseTables (String database) throws DatabaseException
- static ArrayList< String > splitSQL (String sql)
- static String uncommentSQL (String sql)
- static boolean useDatabase (String database) throws DatabaseException

1.6.8.2 Dokumentation der Elementfunktionen

static boolean createDatabase (String database) throws DatabaseException
[static]

Erstellt eine Datenbankschema.

Parameter:

database String Namen des Schema

Rückgabe:

true, wenn erfolgreich

Ausnahmebehandlung:

DatabaseException

Definiert in Zeile 319 der Datei Database.java.

Benutzt Database.execute().

static boolean deleteDatabase (String database) throws DatabaseException
[static]

Löscht ein Datenbankschema.

Parameter:

database String Name des Schema , wenn erfolgreich

Ausnahmebehandlung:

DatabaseException

Definiert in Zeile 332 der Datei Database.java.

Benutzt Database.execute().

static synchronized ArrayList<Integer> execute (String sql) throws DatabaseException [static]

Führt eine oder mehrere durch Simikolon getrennte SQL Anfragen aus und liefert ein Array mit Rückgaben für jede Anfrage.

Folgende Werte kann das Array enthalten: -1 die Anfrage war nicht erfolgreich (gibt es nicht wirklich. Es werden vorher Exceptions geworfen) 0 die Anfrage war erfolgreich, erzeugte aber kein auto_inc Wert >0 die Anfrage erzeugte einen neuen Datensatz mit der ID als Rückgabewert.

Nutzung für DDL, INSERT, UPDATE und DELETE.

Parameter:

sql SQL Anfrage(n) getrennt durch Semikolon.

Ausnahmebehandlung:

DatabaseException

Rückgabe:

-1 Anfrage nicht erfolgreich, 0 Anfrage erfolgreich, > 0 ID eines neuen Datensatzs

Definiert in Zeile 101 der Datei Database.java.

Benutzt Database.splitSQL() und Database.uncommentSQL().

Wird benutzt von Database.createDatabase(), Database.deleteDatabase() und Database.useDatabase().

static boolean existsTable (String table) throws DatabaseException [static]

Gibt zurück, ob eine Tabelle existiert oder nicht.

Parameter:

table String Name der Tabelle

Rückgabe:

true, wenn die Tabelle existiert

Ausnahmebehandlung:

DatabaseException

Definiert in Zeile 388 der Datei Database.java.

static List<String> getDatabaseTables (String database) throws DatabaseException
[static]

Liefert ein ResultSet zurück, welches alle Tabellen eines Schema enthält.

Parameter:

database String Name des Schema

Rückgabe:

ResultSet Liste von Tabellen

Ausnahmebehandlung:

DatabaseException

Definiert in Zeile 359 der Datei Database.java.

static ArrayList<String> splitSQL (String sql) [static]

Trennt eine gegebene SQL Zeichenfolge an den Semikolon (;). Werden mehrere SQL-Anfragen gestellt, so sind diese durch Semikolon, gefolgt von einem Umbruch zu schreiben.

Parameter:

sql String

Rückgabe:

List<String>

Definiert in Zeile 43 der Datei Database.java.

Wird benutzt von Database.execute().

static String uncommentSQL (String sql) [static]

Entfernt SQL Kommentare beginnend mit "--" aus dem angegebenen SQL Skript.

Parameter:

sql String SQL Skript

Rückgabe:

String SQL Skript ohne Kommentare (–)

Definiert in Zeile 77 der Datei Database.java.

Wird benutzt von Database.execute().

static boolean useDatabase (String database) throws DatabaseException [static]

Wählt das Standard-Schema der Datenbank aus.

Parameter:

database String Name des Schema

Rückgabe:

true, wenn das Schema existiert

Ausnahmebehandlung:

DatabaseException

Definiert in Zeile 345 der Datei Database.java.

Benutzt Database.execute().

1.6.9 Column Klassenreferenz

1.6.9.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei Column.java.

1.7 hibernate

1.7.1 HibernateJPAUtil Klassenreferenz

1.7.1.1 Ausführliche Beschreibung

Zentrale Klasse, die Hibernate (anhand der Annotationen) konfiguriert Darüberhinaus werden hier weitere Konfigurationsparameter gesetzt

Definiert in Zeile 17 der Datei HibernateJPAUtil.java.

Öffentliche, statische Methoden

- static EntityManagerFactory getSessionFactory ()

1.7.1.2 Dokumentation der Elementfunktionen

static EntityManagerFactory getSessionFactory () [static]

Rückgabe:

Zentrale Entitymanagerfactory

Definiert in Zeile 109 der Datei HibernateJPAUtil.java.

1.7.2 HibernateTransactionServletfilter Klassenreferenz

Abgeleitet von javax::servlet::Filter.

1.7.2.1 Ausführliche Beschreibung

Servletfilter um den DAOs zu ermöglichen innerhalb einer Transaktion zu arbeiten

Definiert in Zeile 22 der Datei HibernateTransactionServletfilter.java.

Öffentliche Methoden

- void beginTransaction ()
- void commitTransaction ()
- void rollbackTransaction ()

1.7.2.2 Dokumentation der Elementfunktionen

void beginTransaction ()

Startet eine Transaktion manuell **Diese Methode ist nur in Ausnahmefällen zu verwenden**

Definiert in Zeile 71 der Datei HibernateTransactionServletfilter.java.

void commitTransaction ()

Führt ein commit der laufenden Transaktion manuell durch **Diese Methode ist nur in Ausnahmefällen zu verwenden**

Definiert in Zeile 79 der Datei HibernateTransactionServletfilter.java.

void rollbackTransaction ()

Führt ein Rollback der laufenden Transaktion manuell durch **Diese Methode ist nur in Ausnahmefällen zu verwenden**

Definiert in Zeile 87 der Datei HibernateTransactionServletfilter.java.

1.7.3 Setup Klassenreferenz

1.7.3.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 11 der Datei hibernate/Setup.java.

1.8 import_export

1.8.1 ErrorHandler Klassenreferenz

1.8.1.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 13 der Datei import_export/ErrorHandler.java.

1.8.2 ExportAttributeDataTypeMissingException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::import_export::exceptions::ExportException.

1.8.2.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei ExportAttributeDataTypeMissingException.java.

1.8.3 ExportAttributeTypeMissingException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::import_export::exceptions::ExportException.

1.8.3.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei ExportAttributeTypeMissingException.java.

1.8.4 ExportException Klassenreferenz

Basisklasse für ExportAttributeDataTypeMissingException, ExportAttributeTypeMissingException, ExportFirstRelationMemberMissingException und ExportSecondRelationMemberMissingException.

1.8.4.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 9 der Datei ExportException.java.

1.8.5 ExportFirstRelationMemberMissingException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::import_export::exceptions::ExportException.

1.8.5.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei ExportFirstRelationMemberMissingException.java.

1.8.6 ExportSecondRelationMemberMissingException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::import_export::exceptions::ExportException.

1.8.6.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei ExportSecondRelationMemberMissingException.java.

1.8.7 ImportError Klassenreferenz

Basisklasse für ImportInstancesException, ImportIOException, ImportMetamodelException und ImportParseException.

1.8.7.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 9 der Datei ImportError.java.

1.8.8 ImportInstancesException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::import_export::exceptions::ImportException.

1.8.8.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 9 der Datei ImportInstancesException.java.

1.8.9 ImportIOException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::import_export::exceptions::ImportException.

1.8.9.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 9 der Datei ImportIOException.java.

1.8.10 ImportMetamodelException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::import_export::exceptions::ImportException.

1.8.10.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 9 der Datei ImportMetamodelException.java.

1.8.11 ImportParseException Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::import_export::exceptions::ImportException`.

1.8.11.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 9 der Datei `ImportParseException.java`.

1.8.12 Export Klassenreferenz

1.8.12.1 Ausführliche Beschreibung

Diese Klasse ermöglicht das Exportieren von Metamodellen und Instanzen aus dem Kern.

Autor:

Roland Koppe

Definiert in Zeile 17 der Datei `Export.java`.

Öffentliche Methoden

- `void addExportElement (Model exportModel)`
- `boolean export (boolean validateExport)`
- `Export (String version, Date date, String destMetamodel, String destInstances, String refMetamodel)`
- `Export (String version, Date date, String destMetamodel, String destInstances)`

1.8.12.2 Beschreibung der Konstruktoren und Destruktoren

Export (String version, Date date, String destMetamodel, String destInstances)

Konstruktor mit der Angabe von Informationen für den Export. Sollen nur Metamodelle exportiert werden, so kann `destInstances` freigelassen werden ("" oder null). Das Exportieren von Instanzen erfordert hin- gegen auch das Exportieren von dazugehörigen Metamodellen, da sich die Instanzen auf diese Metamodell-Informationen beziehen.

Parameter:

version String Version des Exports (durch Benutzer festgelegt)

date Date Datum des Exports (durch Benutzer festgelegt)

destMetamodel String Ziel des Metamodells

destInstances String Ziel von Instanzen

Definiert in Zeile 41 der Datei Export.java.

Export (String version, Date date, String destMetamodel, String destInstances, String refMetamodel)

Konstruktor mit der Angabe von Informationen für den Export. Sollen nur Metamodelle exportiert werden, so kann destInstances freigelassen werden ("" oder null). Das Exportieren von Instanzen erfordert hingegen auch das Exportieren von dazugehörigen Metamodellen, da sich die Instanzen auf diese Metamodell-Informationen beziehen.

Parameter:

version String Version des Exports (durch Benutzer festgelegt)

date Date Datum des Exports (durch Benutzer festgelegt)

destMetamodel String Ziel des Metamodells

destInstances String Ziel von Instanzen

refMetamodel String Name der Referenz von Instanzen auf das Metamodell

Definiert in Zeile 62 der Datei Export.java.

1.8.12.3 Dokumentation der Elementfunktionen

void addExportElement (Model exportModel)

Fügt zu exportierende Metamodell hinzu.

Parameter:

exportModel ExportModel

Definiert in Zeile 77 der Datei Export.java.

Wird benutzt von ImportExportBean.exportMetamodel().

boolean export (boolean validateExport)

Exportiert die mit addExportElement() angegebenen Metamodelle und wenn im Konstruktor angegeben auch die dazugehörigen Instanzen.

Parameter:

validateExport boolean, wenn true, werden die erzeugten XML Dokumente im Anschluss validiert.

Rückgabe:

true, wenn erfolgreich

Definiert in Zeile 92 der Datei Export.java.

Benutzt ExportInstances.export() und ExportMetamodel.export().

Wird benutzt von ImportExportBean.exportMetamodel().

1.8.13 ExportInstances Klassenreferenz

1.8.13.1 Ausführliche Beschreibung

Diese Klasse ermöglicht das Exportieren von Instanzen.

Autor:

Roland Koppe

Definiert in Zeile 33 der Datei ExportInstances.java.

Öffentliche Methoden

- `boolean export (String version, Date date, String destination, String metamodelURL, ArrayList< Model > exportModels, boolean validateExport)`

1.8.13.2 Dokumentation der Elementfunktionen

boolean export (String version, Date date, String destination, String metamodelURL, ArrayList< Model > exportModels, boolean validateExport)

Exportiert die Instanzen zu den angegebenen Metamodellen.

Parameter:

version String Version durch den Benutzer festgelegt

date Date Datum durch den Benutzer festgelegt

destination String Ziel des XML Dokuments

metamodelURL String Quelle des dazugehörigen Metamodell XML Dokuments

exportModels ArrayList<ExportModel> zu exportierende Metamodelle

validateExport boolean, true wenn erzeugtes XML Dokument validiert werden soll

Rückgabe:

true, wenn erfolgreich

Definiert in Zeile 57 der Datei ExportInstances.java.

Wird benutzt von Export.export().

1.8.14 ExportMetamodel Klassenreferenz

1.8.14.1 Ausführliche Beschreibung

Diese Klasse ermöglicht das Exportieren von Metamodellen.

Autor:

Roland Koppe

Definiert in Zeile 38 der Datei ExportMetamodel.java.

Öffentliche Methoden

- boolean export (String version, Date date, String destination, ArrayList< Model > exportModels, boolean validateExport)

1.8.14.2 Dokumentation der Elementfunktionen

boolean export (String version, Date date, String destination, ArrayList< Model > exportModels, boolean validateExport)

Exportiert die angegebenen Metamodelle.

Parameter:

version String Version durch den Benutzer festgelegt

date Date Datum durch den Benutzer festgelegt

destination String Ziel des XML Dokuments

exportModels ArrayList<ExportModel> zu exportierende Metamodelle

validateExport boolean, true wenn erzeugtes XML Dokument validiert werden soll

Rückgabe:

true, wenn erfolgreich

Definiert in Zeile 60 der Datei ExportMetamodel.java.

Wird benutzt von Export.export().

1.8.15 AnalyseInstances Klassenreferenz

1.8.15.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 28 der Datei AnalyseInstances.java.

1.8.16 AnalyseInstancesContent Klassenreferenz

Abgeleitet von org::xml::sax::ContentHandler.

1.8.16.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 34 der Datei AnalyseInstancesContent.java.

1.8.17 AnalyseMetamodel Klassenreferenz

1.8.17.1 Ausführliche Beschreibung

Analysiert das in einem XML Dokument angegebenen Metamodell.

Autor:

Roland Koppe

Definiert in Zeile 55 der Datei AnalyseMetamodel.java.

1.8.18 AnalyseMetamodelContent Klassenreferenz

Abgeleitet von org::xml::sax::ContentHandler.

1.8.18.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 30 der Datei AnalyseMetamodelContent.java.

1.8.19 Import Klassenreferenz

1.8.19.1 Ausführliche Beschreibung

Diese Klasse ermöglicht das Importieren von Metamodellen und Instanzen aus XML Dokumenten in den Kern.

Autor:

Roland Koppe

Definiert in Zeile 36 der Datei Import.java.

1.8.20 ObjectInstance Klassenreferenz

1.8.20.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei ObjectInstance.java.

1.8.21 ObjectInstances Klassenreferenz

1.8.21.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei ObjectInstances.java.

1.8.22 RelationInstance Klassenreferenz

1.8.22.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei RelationInstance.java.

1.8.23 RelationInstances Klassenreferenz

1.8.23.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei RelationInstances.java.

1.8.24 IEDefinition Klassenreferenz

1.8.24.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 11 der Datei IEDefinition.java.

1.8.25 IEInstancesDocument Klassenreferenz

1.8.25.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei IEInstancesDocument.java.

1.8.26 IEMetamodelDocument Klassenreferenz

1.8.26.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 14 der Datei IEMetamodelDocument.java.

1.8.27 ExportModel Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::import_export::model::Model.

1.8.27.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 21 der Datei ExportModel.java.

Öffentliche Methoden

- ExportModel (EAMCategory cat, boolean exportInstances)
- ExportModel (long metamodelId, boolean exportInstances)

1.8.27.2 Beschreibung der Konstruktoren und Destruktoren

ExportModel (long metamodelId, boolean exportInstances)

Konstruktor wählt das angegebene Metamodell und aggregiert die für den Export notwendigen Informationen.

Parameter:

metamodelId long Id des Metamodells

exportInstances boolean, true für Export von Instanzen

Definiert in Zeile 32 der Datei ExportModel.java.

ExportModel (EAMCategory cat, boolean exportInstances)

Konstruktor wählt das angegebene Metamodell und aggregiert die für den Export notwendigen Informationen.

Parameter:

cat Category Metamodell

exportInstances boolean, true für Export von Instanzen

Definiert in Zeile 48 der Datei ExportModel.java.

1.8.28 IEAttribute Klassenreferenz

1.8.28.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei IEAttribute.java.

1.8.29 IEAttributeType Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::import_export::model::IEType`.

1.8.29.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei IEAttributeType.java.

1.8.30 IEObject Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::import_export::model::IESuperClass`.

1.8.30.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei IEObject.java.

1.8.31 IEObjectType Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::import_export::model::IEType`.

1.8.31.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei IEObjectType.java.

1.8.32 IERelation Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::import_export::model::IESuperClass`.

1.8.32.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei `IERelation.java`.

1.8.33 IERelationType Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::import_export::model::IEType`.

1.8.33.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei `IERelationType.java`.

1.8.34 IEStatus Klassenreferenz

1.8.34.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei `IEStatus.java`.

1.8.35 IESuperClass Klassenreferenz

Basisklasse für `IEObject` und `IERelation`.

1.8.35.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei `IESuperClass.java`.

1.8.36 IETime Klassenreferenz

1.8.36.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei IETime.java.

1.8.37 IEType Klassenreferenz

Basisklasse für IEAttributeType, IEObjectType und IERelationType.

1.8.37.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei IEType.java.

1.8.38 ImportModel Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::import_export::model::Model.

1.8.38.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei ImportModel.java.

1.8.39 Model Klassenreferenz

Basisklasse für ExportModel und ImportModel.

1.8.39.1 Ausführliche Beschreibung

Diese Klasse definiert die für den Export ausgewählten Metamodelle.

Autor:

Roland Koppe

Definiert in Zeile 23 der Datei Model.java.

Öffentliche Methoden

- `ArrayList< EAMAttributeType > getAttributeTypes ()`
- `String getDescription ()`
- `long getId ()`
- `String getName ()`
- `ArrayList< EAMObject > getObjects ()`
- `ArrayList< EAMObjectType > getObjectTypes ()`
- `ArrayList< EAMRelation > getRelations ()`
- `ArrayList< EAMRelationType > getRelationTypes ()`
- `ArrayList< EAMStatus > getStatuss ()`
- `ArrayList< EAMTime > getTimes ()`
- `boolean isExportInstances ()`

1.8.39.2 Dokumentation der Elementfunktionen

`ArrayList<EAMAttributeType> getAttributeTypes ()`

Liefert die EAMAttributeTypes des Metamodells.

Rückgabe:

`ArrayList<EAMAttributeType>`

Definiert in Zeile 213 der Datei Model.java.

`String getDescription ()`

Liefert die Beschreibung des Metamodells.

Rückgabe:

`String` Beschreibung des Metamodells.

Definiert in Zeile 79 der Datei Model.java.

`long getId ()`

Liefert die Id des Metamodells.

Rückgabe:

`long` Id des Metamodells.

Definiert in Zeile 51 der Datei Model.java.

`String getName ()`

Liefert den Namen des Metamodells.

Rückgabe:

String Name des Metamodells.

Definiert in Zeile 65 der Datei Model.java.

ArrayList<EAMObject> getObjects ()

Liefert die EAMObjects des Metamodells.

Rückgabe:

ArrayList<EAMObject>

Definiert in Zeile 116 der Datei Model.java.

ArrayList<EAMObjectType> getObjectTypes ()

Liefert die EAMObjectTypes des Metamodells.

Rückgabe:

ArrayList<EAMObjectType>

Definiert in Zeile 165 der Datei Model.java.

ArrayList<EAMRelation> getRelations ()

Liefert die EAMRelations des Metamodells.

Rückgabe:

ArrayList<EAMRelation>

Definiert in Zeile 140 der Datei Model.java.

ArrayList<EAMRelationType> getRelationTypes ()

Liefert die EAMRelationTypes des Metamodells.

Rückgabe:

ArrayList<EAMRelationType>

Definiert in Zeile 189 der Datei Model.java.

ArrayList<EAMStatus> getStatus ()

Liefert die Status des Metamodells.

Rückgabe:

ArrayList<Status>

Definiert in Zeile 261 der Datei Model.java.

ArrayList<EAMTime> getTimes ()

Liefert die Times des Metamodells.

Rückgabe:

ArrayList<Time>

Definiert in Zeile 237 der Datei Model.java.

boolean isExportInstances ()

Liefert, ob Instanzen exportiert werden sollen.

Rückgabe:

boolean true Instanzen sollen exportiert werden

Definiert in Zeile 93 der Datei Model.java.

1.9 logging

1.9.1 ILoggingDAO Schnittstellenreferenz

1.9.1.1 Ausführliche Beschreibung

DAO für den Zugriff auf die Log-Dateien des EAM-Tools.

Autor:

Roland Koppe

Definiert in Zeile 15 der Datei ILoggingDAO.java.

1.9.2 Logger Klassenreferenz

1.9.2.1 Ausführliche Beschreibung

Rudimentäres Logging in der Konsole oder in eine Datei.

Autor:

Roland Koppe

Definiert in Zeile 16 der Datei Logger.java.

Öffentliche, statische Methoden

- static void close ()
- static void fatal (String info)
- static synchronized Logger getLogger ()
- static LogModes getLogMode ()
- static String getLogPath ()
- static void info (String info)
- static boolean isConsoleTime ()
- static void setConsoleTime (boolean displayTime)
- static void setLogMode (LogModes mode)
- static void setLogPath (String logPath)
- static void warning (String info)

1.9.2.2 Dokumentation der Elementfunktionen

static void close () [static]

Beendet das Logging (END).

Definiert in Zeile 218 der Datei Logger.java.

static void fatal (String info) [static]

Loggt die angegebene Information als (FATAL).

Parameter:

info String

Definiert in Zeile 198 der Datei Logger.java.

static synchronized Logger getLogger () [static]

Liefert das Logging-Objekt.

Rückgabe:

Log Logging-Objekt

Definiert in Zeile 46 der Datei Logger.java.

static LogModes getLogMode () [static]

Liefert den aktuellen Log-Modus.

Rückgabe:

LogModes

Definiert in Zeile 60 der Datei Logger.java.

static String getLogPath () [static]

Liefert den Namen des Log-Pfads.

Rückgabe:

String Name des Log-Pfads

Definiert in Zeile 100 der Datei Logger.java.

static void info (String info) [static]

Loggt die angegebene Information als (INFO).

Parameter:

info String

Definiert in Zeile 187 der Datei Logger.java.

static boolean isConsoleTime () [static]

Liefert zurück, ob in der Konsole die aktuelle Zeit ausgegeben wird.

Rückgabe:

true, wenn Konsolenausgabe mit Zeit

Definiert in Zeile 80 der Datei Logger.java.

static void setConsoleTime (boolean displayTime) [static]

Setzt, ob in der Konsole die aktuelle Zeit ausgegeben werden soll.

Parameter:

displayTime boolean

Definiert in Zeile 89 der Datei Logger.java.

static void setLogMode (LogModes mode) [static]

Setzt den Log-Modus.

Parameter:

mode LogModes

Definiert in Zeile 69 der Datei Logger.java.

static void setLogPath (String logPath) [static]

Setzt den Pfad für das Logging.

Parameter:

logPath String Pfad

Definiert in Zeile 109 der Datei Logger.java.

static void warning (String info) [static]

Loggt die angegebene Information als (WARNING).

Parameter:

info String

Definiert in Zeile 209 der Datei Logger.java.

1.9.3 LoggingDAO Klassenreferenz

1.9.3.1 Ausführliche Beschreibung

DAO für den Zugriff auf die Log-Dateien des EAM-Tools.

Autor:

Roland Koppe

Definiert in Zeile 24 der Datei LoggingDAO.java.

1.9.4 Log Klassenreferenz

1.9.4.1 Ausführliche Beschreibung

Diese Klasse repräsentiert den Verweis auf eine Log-Datei.

Autor:

Roland Koppe

Definiert in Zeile 11 der Datei Log.java.

1.10 menu

1.10.1 AbstractAdoptableMenuContributor Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::api::AbstractMenuContributor und de::offis::pg::eam::core::menu::api::IMenuContributorAdopter.

Basisklasse für TopMenu.

1.10.1.1 Ausführliche Beschreibung

Im Vergleich zu `AbstractMenuContributor` und `AbstractMenuAdopter` wird diese Klasse benutzt im Fall, dass eine Klasse sowohl Menü beiträgt auch Menü adoptiert.

Autor:

Yu

Definiert in Zeile 18 der Datei `AbstractAdoptableMenuContributor.java`.

Öffentliche Methoden

- `void addMenuContributor (IMenuContributor menuContributor)`
- `final LogService getLog ()`
- `final List< IMenuContributor > getMenuContributors ()`
- `void removeMenuContributor (IMenuContributor menuContributor)`
- `final void setLog (LogService log)`
- `final void unsetLog (LogService log)`

Geschützte Methoden

- `IMenuItem locateMenuItem (IMenuItem item, List< String > pfad, int i)`

1.10.1.2 Dokumentation der Elementfunktionen

`void addMenuContributor (IMenuContributor menuContributor)`

Fügt einen Contributor hinzu.

Implementiert `IMenuContributorAdopter`.

Erneute Implementation in `TopMenu`.

Definiert in Zeile 42 der Datei `AbstractAdoptableMenuContributor.java`.

Benutzt `IMenuContributorAdopter.getAdopterId()`, `Submenu.getChildren()`, `IMenuContributor.getContributionTarget()`, `IMenuContributor.getContributorId()`, `AbstractAdoptableMenuContributor.getMenuContributors()`, `AbstractMenuContributor.getMenuItems()`, `IMenuContributor.getTargetSubmenu()` und `AbstractAdoptableMenuContributor.locateMenuItem()`.

`final LogService getLog ()`

Liefert den LogService zurück.

Rückgabe:

Definiert in Zeile 179 der Datei AbstractAdoptableMenuContributor.java.

final List<IMenuContributor> getMenuContributors ()

Liefert eine Liste von MenuContribuors zurück.

Implementiert IMenuContributorAdopter.

Definiert in Zeile 35 der Datei AbstractAdoptableMenuContributor.java.

Wird benutzt von AbstractAdoptableMenuContributor.addMenuContributor() und AbstractAdoptableMenuContributor.removeMenuContributor().

MenuItem locateMenuItem (MenuItem item, List< String > pfd, int i) [protected]

Liefert das Menü zurück, wo das als Parameter übergebene MenuItem landen soll. Und Zwar, diese Methode lokalisiert, in Welches Menü soll ein beigetragenes Menü eingesteckt werden.

Parameter:

item der zu lokalisierende Menüeintrag

pfd gibt an, wo soll ein Menüeintrag hinzugefügt wird.

i gibt die Stufe an.

Rückgabe:

der Ort, wo das als Parameter übergebene Menü hinzugefügt wird.

Definiert in Zeile 92 der Datei AbstractAdoptableMenuContributor.java.

Benutzt Submenu.getChildren(), MenuItem.getId() und MenuItem.getType().

Wird benutzt von AbstractAdoptableMenuContributor.addMenuContributor() und AbstractAdoptableMenuContributor.removeMenuContributor().

void removeMenuContributor (IMenuContributor menuContributor)

Entfernt ein MenuContriburo.

Implementiert IMenuContributorAdopter.

Erneute Implementation in TopMenu.

Definiert in Zeile 115 der Datei AbstractAdoptableMenuContributor.java.

Benutzt IMenuContributorAdopter.getAdopterId(), Submenu.getChildren(), IMenuContributor.getContributionTarget(), IMenuContributor.getContributorId(), AbstractAdoptableMenuContributor.getMenuContributors(),

`AbstractMenuContributor.getItems()`, `IMenuContributor.getTargetSubmenu()` und `AbstractAdoptableMenuContributor.locateMenuItem()`.

final void setLog (LogService log)

Setter des LogService

Parameter:

log

Definiert in Zeile 170 der Datei `AbstractAdoptableMenuContributor.java`.

final void unsetLog (LogService log)

Entfernt einen LogService

Parameter:

log

Definiert in Zeile 158 der Datei `AbstractAdoptableMenuContributor.java`.

1.10.2 AbstractMenuContributor Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::menu::api::IMenuContributor`.

Basisklasse für `AbstractAdoptableMenuContributor`.

1.10.2.1 Ausführliche Beschreibung

Diese Klasse bietet defaulte Implementierung der Schnittstelle `IMenuContributor` und erlaubt auch die Initialisierung eines Menüs mit Hilfe einer Konfigurationsdatei `menu_config.xml`. Diese Datei muss sich mit dem Contributor im gleichen Verzeichnis befinden.

Autor:

Yu

Definiert in Zeile 19 der Datei `AbstractMenuContributor.java`.

Öffentliche Methoden

- `final boolean addItem (IMenuItem item)`
- `final void addItem (Set< IMenuItem > items)`
- `final Set< IMenuItem > getMenuItems ()`
- `final String getSymbolicName ()`
- `int hashCode ()`

Geschützte Methoden

- void initMenuContributorFromDefaultConfigXML ()

1.10.2.2 Dokumentation der Elementfunktionen

final boolean addMenuItem (IMenuItem item)

Fügt ein Menüeintrag hinzu.

Parameter:

item ein neuer Menüeintrag

Rückgabe:

true, falls der Menüeintrag erfolgreich hinzugefügt wird.

Definiert in Zeile 39 der Datei AbstractMenuContributor.java.

final void addMenuItems (Set< IMenuItem > items)

Setter der Menge der Menüeinträge

Parameter:

items

Definiert in Zeile 48 der Datei AbstractMenuContributor.java.

Wird benutzt von

AbstractMenuContributor.initMenuContributorFromDefaultConfigXML().

final Set<IMenuItem> getMenuItems ()

Liefert die Menge der Menüeinträge zurück.

Implementiert IMenuContributor.

Definiert in Zeile 55 der Datei AbstractMenuContributor.java.

Wird benutzt von AbstractAdoptableMenuContributor.addMenuContributor(),
TopMenu.getTopMenuParams() und
AbstractAdoptableMenuContributor.removeMenuContributor().

final String getSymbolicName ()

Liefert den symbolischen Namen zurück.

Implementiert IMenuContributor.

Definiert in Zeile 63 der Datei AbstractMenuContributor.java.

int hashCode ()

hashCode

Definiert in Zeile 71 der Datei AbstractMenuContributor.java.

void initMenuContributorFromDefaultConfigXML () [protected]

initialize ein Menü mit Hilfe einer Konfigurationsdatei menu_config.xml, die sich mit dem Contributor im vergleichen Verzeichnis befindet.

Definiert in Zeile 79 der Datei AbstractMenuContributor.java.

Benutzt AbstractMenuContributor.addMenuItems().

Wird benutzt von TopMenu.TopMenu().

1.10.3 AbstractMenuContributorAdopter Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::api::IMenuContributorAdopter.

Basisklasse für TopMenuBar.

1.10.3.1 Ausführliche Beschreibung

Diese Klasse wird benutzt, um MenuContributor zu adoptieren.

Autor:

Yu

Definiert in Zeile 15 der Datei AbstractMenuContributorAdopter.java.

Öffentliche Methoden

- final void addMenuContributor (IMenuContributor menuContributor)
- final LogService getLog ()
- final List< IMenuContributor > getMenuContributors ()
- final void removeMenuContributor (IMenuContributor menuContributor)
- final void setLog (LogService log)
- final void unsetLog (LogService log)

1.10.3.2 Dokumentation der Elementfunktionen**final void addMenuContributor (IMenuContributor menuContributor)**

Fügt ein MenuContributor hinzu.

Implementiert IMenuContributorAdopter.

Definiert in Zeile 39 der Datei AbstractMenuContributorAdopter.java.

Benutzt IMenuContributorAdopter.getAdopterId(),
IMenuContributor.getContributionTarget() und IMenuContributor.getContributorId().

final LogService getLog ()

Liefert den LogService zurück.

Rückgabe:

Definiert in Zeile 88 der Datei AbstractMenuContributorAdopter.java.

final List<IMenuContributor> getMenuContributors ()

Liefert die Liste von MenuContributors zurück.

Implementiert IMenuContributorAdopter.

Definiert in Zeile 32 der Datei AbstractMenuContributorAdopter.java.

final void removeMenuContributor (IMenuContributor menuContributor)

Entfernt ein MenuContributor

Implementiert IMenuContributorAdopter.

Definiert in Zeile 53 der Datei AbstractMenuContributorAdopter.java.

Benutzt IMenuContributorAdopter.getAdopterId(),
IMenuContributor.getContributionTarget() und IMenuContributor.getContributorId().

final void setLog (LogService log)

Setter des LogService

Parameter:

log

Definiert in Zeile 80 der Datei AbstractMenuContributorAdopter.java.

final void unsetLog (LogService log)

Stoppt einen log service

Parameter:

log ein neuer LogService

Definiert in Zeile 69 der Datei AbstractMenuContributorAdopter.java.

1.10.4 AuthMenuItem Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::menu::api::IMenuItem` und `de::offis::pg::eam::core::menu::impl::IAccessRight`.

1.10.4.1 Ausführliche Beschreibung

Diese Klasse wird benutzt, um die autorisierten Menueinträgen zu schützen. Wenn ein Nutzer das Recht hat, einen Menüeintrag zu sehen. Dann wird die Generierung der Menüparameter zum geschützten Menüeintrag delegiert, sonst wird die Generierung verweigert.

Autor:

Yu

Definiert in Zeile 18 der Datei `AuthMenuItem.java`.

Öffentliche Methoden

- `void accept (IMenuVisitor visitor)`
- `AuthMenuItem (IMenuItem delegate, String view)`
- `int generateMenuParams (int dynId, Map< String, String > newParams)`
- `String getBundleName ()`
- `List< IMenuItem > getChildren ()`
- `IMenuItem getDelegate ()`
- `String getId ()`
- `String getParam (String key)`
- `Map< String, String > getParams ()`
- `ItemType getType ()`
- `String getView ()`
- `void setBundleName (String bundleName)`
- `void setDelegate (IMenuItem delegate)`
- `void setParam (String key, String value)`
- `void setView (String right)`
- `String toString ()`

1.10.4.2 Beschreibung der Konstruktoren und Destruktoren

`AuthMenuItem (IMenuItem delegate, String view)`

Konstruktor

Parameter:

delegate das zu schützende Menü

view der View eines Menüs

Definiert in Zeile 46 der Datei AuthMenuItem.java.

1.10.4.3 Dokumentation der Elementfunktionen

void accept (IMenuVisitor visitor)

akzeptiert einen Menüvisitor

Implementiert IMenuItem.

Definiert in Zeile 103 der Datei AuthMenuItem.java.

Benutzt IMenuVisitor.visit().

int generateMenuParams (int dynld, Map< String, String > newParams)

diese Methode schützt die Generierung der Menüparameter eines Menüs.

Implementiert IMenuItem.

Definiert in Zeile 55 der Datei AuthMenuItem.java.

Benutzt IMenuItem.generateMenuParams().

String getBundleName ()

Liefert den Bundlenamen zurück.

Implementiert IMenuItem.

Definiert in Zeile 152 der Datei AuthMenuItem.java.

Benutzt IMenuItem.getBundleName().

Wird benutzt von JddmMenuParamsGenerator.visit().

List<IMenuItem> getChildren () [virtual]

Liefert Unterpunkte eines Submenüs zurück.

Implementiert IMenuItem.

Definiert in Zeile 63 der Datei AuthMenuItem.java.

Benutzt IMenuItem.getChildren().

IMenuItem getDelegate ()

Liefert das geschützte MenuItem zurück.

Rückgabe:

Definiert in Zeile 112 der Datei AuthMenuItem.java.

Wird benutzt von JddmMenuParamsGenerator.visit().

String getId ()

Liefert die ID des Menüs zurück.

Implementiert IMenuItem.

Definiert in Zeile 145 der Datei AuthMenuItem.java.

Benutzt IMenuItem.getId().

String getParam (String key) [virtual]

Liefer den Wert eines Parameters zurück.

Implementiert IMenuItem.

Definiert in Zeile 71 der Datei AuthMenuItem.java.

Benutzt IMenuItem.getParam().

Map<String, String> getParams () [virtual]

Liefert das Map der Parameter zurück.

Implementiert IMenuItem.

Definiert in Zeile 79 der Datei AuthMenuItem.java.

Benutzt IMenuItem.getParams().

ItemType getType () [virtual]

Liefert den Typ eines Menü zurück.

Implementiert IMenuItem.

Definiert in Zeile 87 der Datei AuthMenuItem.java.

Benutzt IMenuItem.getType().

String getView ()

Liefert den View zurück.

Implementiert IAccessRight.

Definiert in Zeile 129 der Datei AuthMenuItem.java.

Wird benutzt von JddmMenuParamsGenerator.visit().

void setBundleName (String bundleName)

Setter des Bundlenamens

Implementiert IMenuItem.

Definiert in Zeile 159 der Datei AuthMenuItem.java.

Benutzt IMenuItem.setBundleName().

void setDelegate (IMenuItem delegate)

Setter des geschützten MenuItems

Parameter:

delegate

Definiert in Zeile 121 der Datei AuthMenuItem.java.

void setParam (String key, String value) [virtual]

Setter der Menüparameter

Implementiert IMenuItem.

Definiert in Zeile 95 der Datei AuthMenuItem.java.

Benutzt IMenuItem.setParam().

void setView (String right)

Setter des Views

Implementiert IAccessRight.

Definiert in Zeile 137 der Datei AuthMenuItem.java.

String toString ()

AuthMenuItem als String

Definiert in Zeile 24 der Datei AuthMenuItem.java.

1.10.5 IMenuContributor Schnittstellenreferenz

Basisklasse für AbstractMenuContributor.

1.10.5.1 Ausführliche Beschreibung

Diese Schnittstelle wird dazu benutzt, um Ein Bundle seine Menü am Kern anzumelden.

Autor:

Yu

Definiert in Zeile 12 der Datei IMenuContributor.java.

Öffentliche Methoden

- String getContributionTarget ()
- String getContributorId ()
- Set< IMenuItem > getMenuItems ()
- String getSymbolicName ()
- String getTargetSubmenu ()

1.10.5.2 Dokumentation der Elementfunktionen

String getContributionTarget ()

Liefert einen Pfad im Zielmenü zurück, wo die Liste von Menueinträgen landen sollen.

Rückgabe:

ein Prad im Zielmenü

Implementiert in TopMenu.

Wird benutzt von AbstractMenuContributorAdopter.addMenuContributor(), AbstractAdoptableMenuContributor.addMenuContributor(), AbstractMenuContributorAdopter.removeMenuContributor() und AbstractAdoptableMenuContributor.removeMenuContributor().

String getContributorId ()

Liefert die ID des Contributors zurück.

Rückgabe:

selbe ID

Implementiert in TopMenu.

Wird benutzt von AbstractMenuContributorAdopter.addMenuContributor(), AbstractAdoptableMenuContributor.addMenuContributor(), AbstractMenuContributorAdopter.removeMenuContributor() und AbstractAdoptableMenuContributor.removeMenuContributor().

Set<MenuItem> getMenuItems ()

Liefert eine Menge von Menueinträgen zurück.

Rückgabe:

die beizutragende Menueinträgen

Implementiert in AbstractMenuContributor.

String getSymbolicName ()

Liefert den symbolischen Namen des Bundles zurück.

Rückgabe:

der symbolischen Name des Bundles

Implementiert in AbstractMenuContributor.

String getTargetSubmenu ()

Liefert das Zielmenü zurück.

Rückgabe:

das Zielmenü

Implementiert in TopMenu.

Wird benutzt von AbstractAdoptableMenuContributor.addMenuContributor() und AbstractAdoptableMenuContributor.removeMenuContributor().

1.10.6 IMenuContributorAdopter Schnittstellenreferenz

Basisklasse für AbstractAdoptableMenuContributor und AbstractMenuContributorAdopter.

1.10.6.1 Ausführliche Beschreibung

Diese Klasse wird dazu benutzt, um die beizutragenden Menueinträgen aus anderen Bundles aufzunehmen.

Autor:

Yu

Definiert in Zeile 12 der Datei IMenuContributorAdopter.java.

Öffentliche Methoden

- void addMenuContributor (IMenuContributor menuContributor)

- String getAdopterId ()
- List< IMenuContributor > getMenuContributors ()
- void removeMenuContributor (IMenuContributor menuContributor)

1.10.6.2 Dokumentation der Elementfunktionen

void addMenuContributor (IMenuContributor menuContributor)

Fügt einen MenuContributor hinzu.

Parameter:

menuContributor ein MenuContributor

Implementiert in AbstractAdoptableMenuContributor, AbstractMenuContributorAdopter und TopMenu.

String getAdopterId ()

Liefert selbe ID zurück.

Rückgabe:

selbe ID

Implementiert in TopMenu.

Wird benutzt von AbstractMenuContributorAdopter.addMenuContributor(), AbstractAdoptableMenuContributor.addMenuContributor(), AbstractMenuContributorAdopter.removeMenuContributor() und AbstractAdoptableMenuContributor.removeMenuContributor().

List<IMenuContributor> getMenuContributors ()

Liefert aufgenommene MenuContributors zurück.

Rückgabe:

eine Liste von MenuContributors

Implementiert in AbstractAdoptableMenuContributor und AbstractMenuContributorAdopter.

void removeMenuContributor (IMenuContributor menuContributor)

Entfernt ein MenuContributor.

Parameter:

menuContributor das zu entfernende MenuContributor

Implementiert in AbstractAdoptableMenuContributor, AbstractMenuContributorAdopter und TopMenu.

1.10.7 IMenuItem Schnittstellenreferenz

Basisklasse für AuthMenuItem und MenuItem.

1.10.7.1 Ausführliche Beschreibung

Diese Klasse definiert die Schnittstelle des Menüs und ihre Konstante sowie die Methoden, die ein Menü implementieren muss.

Autor:

Yu

Definiert in Zeile 15 der Datei IMenuItem.java.

Öffentliche Typen

- enum ItemType

Öffentliche Methoden

- void accept (IMenuVisitor visitor)
- int generateMenuParams (int dynId, Map< String, String > newParams)
- String getBundleName ()
- abstract List< IMenuItem > getChildren ()
- String getId ()
- abstract String getParam (String key)
- abstract Map< String, String > getParams ()
- abstract ItemType getType ()
- void setBundleName (String name)
- abstract void setParam (String key, String value)

Statische öffentliche Attribute

- static final String DISCRIPTION = "discription"
- static final String ENABLED = "enabled"
- static final String ID = "id"
- static final String LINK = "link"
- static final String LOCALE_PREFIX = "LOCALE"

- static final String NAME = "name"
- static final String NODE = "node"
- static final String PARAM_FORMATTER = "%s=%s"
- static final String PARAM_SEPERATOR = ";
- static final String SHORTCUT = "shortcut"
- static final String SUBID = "subid"
- static final String TARGET = "target"

1.10.7.2 Dokumentation der Aufzählungstypen

enum ItemType

Konstante, um Submenu und SimpleMenuItem auseinander zu halten.

Autor:

Yu

Definiert in Zeile 90 der Datei IMenuItem.java.

1.10.7.3 Dokumentation der Elementfunktionen

void accept (IMenuVisitor visitor)

Nimmt einen Visitor an.

Parameter:

visitor ein MenuVisitor

Implementiert in AuthMenuItem, SimpleMenuItem und Submenu.

Wird benutzt von JddmMenuParamsGenerator.visit().

int generateMenuParams (int dynId, Map< String, String > newParams)

Die Methode wird dazu benutzt, um dynamische Parameter zu erzeugen, die das JDDM-Menu-Applet richtig verstehen kann.

Parameter:

dynId dynamische ID

newParams ein Map von Parameter

Rückgabe:

die nächste dynamische ID

Implementiert in AuthMenuItem, MenuItem und Submenu.

Wird benutzt von AuthMenuItem.generateMenuParams().

String getBundleName ()

Liefert den Bundlenamen zurück, wozu ein Menüeintrag gehört.

Rückgabe:

der Bundlename

Implementiert in AuthMenuItem und MenuItem.

Wird benutzt von AuthMenuItem.getBundleName().

abstract List<IMenuItem> getChildren () [pure virtual]

Liefert Unterpunkte eines Submenüs zurück.

Rückgabe:

Implementiert in AuthMenuItem, MenuItem und Submenu.

Wird benutzt von AuthMenuItem.getChildren().

String getId ()

Liefert die ID eines Menüeintrags zurück.

Rückgabe:

die ID eines Menüeintrags

Implementiert in AuthMenuItem und MenuItem.

Wird benutzt von AuthMenuItem.getId(),
AbstractAdoptableMenuContributor.locateMenuItem() und
JddmMenuParamsGenerator.visit().

abstract String getParam (String key) [pure virtual]

Liefert den Wert eines Schlüssels zurück

Parameter:

key der Parameterschlüssel

Rückgabe:

der Wert eines Parameterschlüssels

Implementiert in AuthMenuItem und MenuItem.

Wird benutzt von AuthMenuItem.getParam().

abstract Map<String, String> getParams () [pure virtual]

Liefert alle Parameter eines Menüeintrags zurück.

Rückgabe:

alle Parameter eines Menüeintrags

Implementiert in AuthMenuItem und MenuItem.

Wird benutzt von AuthMenuItem.getParams() und MenuItem.MenuItem().

abstract ItemType getType () [pure virtual]

Liefert den Typ eines Menüeintrags zurück.

Rückgabe:

Implementiert in AuthMenuItem, SimpleMenuItem und Submenu.

Wird benutzt von AuthMenuItem.getType() und
AbstractAdoptableMenuContributor.locateMenuItem().

void setBundleName (String name)

Setter des Bündelnamens

Parameter:

name der Bundlename

Implementiert in AuthMenuItem und MenuItem.

Wird benutzt von AuthMenuItem.setBundleName().

abstract void setParam (String key, String value) [pure virtual]

Setter eines Menüparameters

Parameter:

key der Schlüssel

value der Wert

Implementiert in AuthMenuItem und MenuItem.

Wird benutzt von AuthMenuItem.setParam().

1.10.7.4 Dokumentation der Datenelemente

final String DISCRIPTION = "discription" [static]

Die Beschreibung eines Menüeintrags wird in der Statusleiste angezeigt, wenn ein Menüeintrag ein Focuse hat.

Definiert in Zeile 28 der Datei IMenuItem.java.

final String ENABLED = "enabled" [static]

Dieser Parameter bestimmt, ob sich ein Menüeintrag auf die Klickaktion reagiert.

Definiert in Zeile 34 der Datei IMenuItem.java.

final String ID = "id" [static]

Die ID eines Menüeintrags

Definiert in Zeile 39 der Datei IMenuItem.java.

final String LINK = "link" [static]

Der Link eines Menüeintrags, wenn das Menüeintrag geklickt wird, dann wird die Webseite angezeigt, worauf der Link verweist.

Definiert in Zeile 45 der Datei IMenuItem.java.

final String LOCALE_PREFIX = "LOCALE" [static]

Präfix für Locale, die für die Internationalisierung verwendet werden.

Definiert in Zeile 82 der Datei IMenuItem.java.

final String NAME = "name" [static]

Der Name eines Menüeintrags

Definiert in Zeile 50 der Datei IMenuItem.java.

final String NODE = "node" [static]

Dieser Parameter gibt an, ob der Menueintrag ein Submenu oder ein Simplemenu ist.

Definiert in Zeile 56 der Datei IMenuItem.java.

final String PARAM_FORMATER = "%s=%s" [static]

In dieser Format werden die Menüparameter eines Menüs serialisiert und dann an das Menüapplet gesendet. Ein Menüparameter setzt sich aus einem Schlüssel und einem Wert zusammen.

Definiert in Zeile 22 der Datei IMenuItem.java.

final String PARAM_SEPERATOR = ";" [static]

die Seperator der Menüparameter

Definiert in Zeile 77 der Datei IMenuItem.java.

final String SHORTCUT = "shortcut" [static]

Shortcut eines Menüeintrags

Definiert in Zeile 61 der Datei IMenuItem.java.

final String SUBID = "subid" [static]

Dieser Parameter gibt die ID eines Submenüs an, wozu dieser Menüeintrag gehört.

Definiert in Zeile 67 der Datei IMenuItem.java.

final String TARGET = "target" [static]

Wo ein Menüeintrag angezeigt wird.

Definiert in Zeile 72 der Datei IMenuItem.java.

1.10.8 ITopMenuConstants Schnittstellenreferenz

Basisklasse für TopMenu.

1.10.8.1 Ausführliche Beschreibung

Diese Schnittstelle definiert Konstanten des Topmenüs.

Autor:

Yu

Definiert in Zeile 9 der Datei ITopMenuConstants.java.

Statische öffentliche Attribute

- final static String MENU_CONTRIBUTOR_ADDED = "MENU_CONTRIBUTOR_ADDED"
- final static String MENU_CONTRIBUTOR_REMOVED = "MENU_CONTRIBUTOR_REMOVED"
- final static String TOP_MENU_ADOPTER_ID = "TOP_MENU_ADOPTER"
- final static String TOP_MENU_CONTRIBUTOR_ID = "TOP_MENU_CONTRIBUTOR"
- final static String TOP_MENU_ID = "topMenu"

- `final static String TOP_MENU_UPDATE_EVENT = "TOP_MENU_UPDATE_EVENT"`

1.10.8.2 Dokumentation der Datenelemente

`final static String MENU_CONTRIBUTOR_ADDED = "MENU_CONTRIBUTOR_ADDED" [static]`

Konstante für das Hinzufügen eines MenuContributor

Definiert in Zeile 14 der Datei ITopMenuConstants.java.

`final static String MENU_CONTRIBUTOR_REMOVED = "MENU_CONTRIBUTOR_REMOVED" [static]`

Konstante für das Entfernen eines MenuContributor

Definiert in Zeile 19 der Datei ITopMenuConstants.java.

`final static String TOP_MENU_ADOPTER_ID = "TOP_MENU_ADOPTER" [static]`

Die ID für Top-Menu-Adopter

Definiert in Zeile 24 der Datei ITopMenuConstants.java.

`final static String TOP_MENU_CONTRIBUTOR_ID = "TOP_MENU_CONTRIBUTOR" [static]`

Die ID für Top-Menu-Contributor

Definiert in Zeile 29 der Datei ITopMenuConstants.java.

`final static String TOP_MENU_ID = "topMenu" [static]`

Die ID des Topmenüs

Definiert in Zeile 34 der Datei ITopMenuConstants.java.

`final static String TOP_MENU_UPDATE_EVENT = "TOP_MENU_UPDATE_EVENT" [static]`

Die Ereignis-ID der Topmenüaktualisierung

Definiert in Zeile 39 der Datei ITopMenuConstants.java.

1.10.9 MenuContributorUtils Klassenreferenz

1.10.9.1 Ausführliche Beschreibung

Diese Klasse ist ein Parser, der eine Konfigurationsdatei des Menüs einliest und erzeugt einen entsprechenden Baum der Menü, die der Konfiguraitonsdatei entspricht.

Autor:

Yu

Definiert in Zeile 23 der Datei MenuContributorUtils.java.

Öffentliche, statische Methoden

- static Set< IMenuItem > initMenuContributorFromXML (Class classz, String fileName) throws JDOMException, IOException

1.10.9.2 Dokumentation der Elementfunktionen

static Set<IMenuItem> initMenuContributorFromXML (Class classz, String fileName) throws JDOMException, IOException [static]

Diese Methode liest eine Konfigurationsdatei des Menüs ein und erzeugt das entsprechende Menü.

Parameter:

classz eine Klasse, die die Konfigurationsdatei einladen kann, die durch das übergebene Parameter fileName angegeben wird.

fileName der Name der Konfigurationsdatei

Rückgabe:

die erzeugte Menüs

Ausnahmebehandlung:

JDOMException falls dieser Parser die Datei nicht richtig parsen kann.

IOException falls classz diese Datei nicht einladen kann.

Definiert in Zeile 148 der Datei MenuContributorUtils.java.

1.10.10 MenuItem Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::api::IMenuItem.

Basisklasse für SimpleMenuItem und Submenu.

1.10.10.1 Ausführliche Beschreibung

Diese Klasse ist eine defaulte Implementierung der Schnittstelle `MenuItem`.

Autor:

Yu

Definiert in Zeile 17 der Datei `MenuItem.java`.

Öffentliche Methoden

- `int generateMenuParams (int dynId, Map< String, String > newParams)`
- `String getBundleName ()`
- `List< MenuItem > getChildren ()`
- `String getId ()`
- `Map< Locale, String > getLocaleName ()`
- `final String getParam (String key)`
- `final Map< String, String > getParams ()`
- `MenuItem (String id, String name, String url, String discription, String shortcut)`
- `MenuItem (String id, String name, String url, String discription)`
- `MenuItem (String id, String name, String url)`
- `MenuItem (String id, String name)`
- `MenuItem (MenuItem another)`
- `void setBundleName (String bundleName)`
- `void setId (String id)`
- `void setLocaleName (Map< Locale, String > localeName)`
- `void setLocaleName (Locale locale, String name)`
- `final void setParam (String key, String value)`
- `String toString ()`

1.10.10.2 Beschreibung der Konstruktoren und Destruktoren

`MenuItem (MenuItem another)`

Kopiekonstruktor

Parameter:

another ein anderer Menüeintrag

Definiert in Zeile 46 der Datei `MenuItem.java`.

Benutzt `MenuItem.getParams()`.

MenuItem (String id, String name)

Konstruktor

Parameter:

id die ID des Menüeintrags

name der Name des Menüeintrags

Definiert in Zeile 58 der Datei MenuItem.java.

MenuItem (String id, String name, String url)

Konstruktor

Parameter:

id die ID des Menüeintrags

name der Name des Menüeintrags

url der Link des Menüeintrags

Definiert in Zeile 72 der Datei MenuItem.java.

MenuItem (String id, String name, String url, String discription)

Konstruktor

Parameter:

id die ID des Menüeintrags

name der Name des Menüeintrags

url der Link der Menüeintrags

discription die Beschreibung des Menüeintrags

Definiert in Zeile 88 der Datei MenuItem.java.

MenuItem (String id, String name, String url, String discription, String shortcut)

Konstruktor

Parameter:

id die ID des Menüeintrags

name der Name des Menüeintrags

url der Link des Menüeintrags

discription dei Beschreibung des Menüeintrags

shortcut das Shortcut des Menüeintrags

Definiert in Zeile 106 der Datei MenuItem.java.

Benutzt MenuItem.setId().

1.10.10.3 Dokumentation der Elementfunktionen

int generateMenuParams (int dynId, Map< String, String > newParams)

Diese Methode generiert die Menüparameter des Menüeintrags

Implementiert IMenuItem.

Erneute Implementation in Submenu.

Definiert in Zeile 121 der Datei MenuItem.java.

String getBundleName ()

Liefert den Bundlenamen zurück.

Implementiert IMenuItem.

Definiert in Zeile 140 der Datei MenuItem.java.

List<IMenuItem> getChildren () [virtual]

Liefert Unterpunkte eines Menüs zurück.

Implementiert IMenuItem.

Erneute Implementation in Submenu.

Definiert in Zeile 147 der Datei MenuItem.java.

String getId ()

Liefer die ID des Menüs zurück.

Implementiert IMenuItem.

Definiert in Zeile 154 der Datei MenuItem.java.

Map<Locale, String> getLocaleName ()

Liefert alle Locale Namen zurück, die zur Internalisierung benutzt werden.

Rückgabe:

ein Map für Locale Namen

Definiert in Zeile 164 der Datei MenuItem.java.

Wird benutzt von JddmMenuParamsGenerator.visit().

final String getParam (String key) [virtual]

Liefert den Wert eines Menüparameter zurück.

Implementiert IMenuItem.

Definiert in Zeile 171 der Datei MenuItem.java.

Wird benutzt von JddmMenuParamsGenerator.visit().

final Map<String, String> getParams () [virtual]

Liefert das Map für die Menüparameter zurück.

Implementiert IMenuItem.

Definiert in Zeile 178 der Datei MenuItem.java.

Wird benutzt von Submenu.Submenu() und JddmMenuParamsGenerator.visit().

void setBundleName (String bundleName)

Setter des Bundlenamens

Implementiert IMenuItem.

Definiert in Zeile 185 der Datei MenuItem.java.

void setId (String id)

Setter der Bundleid

Parameter:

id

Definiert in Zeile 194 der Datei MenuItem.java.

Wird benutzt von MenuItem.MenuItem().

void setLocaleName (Map< Locale, String > localeName)

Setter der lokalen Namen

Parameter:

localeName

Definiert in Zeile 215 der Datei MenuItem.java.

void setLocaleName (Locale locale, String name)

weist dem Menüeintrag einen lokalen Namen zu.

Parameter:

locale z.B. Locale.German

name der Menüname

Definiert in Zeile 206 der Datei MenuItem.java.

final void setParam (String key, String value) [virtual]

Weist einem Parameter einen Wert zu..

Implementiert JMenuItem.

Definiert in Zeile 222 der Datei JMenuItem.java.

String toString ()

als String

Erneute Implementation in Submenu.

Definiert in Zeile 230 der Datei JMenuItem.java.

1.10.11 PageAccessTrackServlet Klassenreferenz

Abgeleitet von javax::servlet::http::HttpServlet.

1.10.11.1 Ausführliche Beschreibung

Diese Klasse wird zur Nutzerverfolgung benutzt und notiert, an welchen Webseiten ein Nutzer gerade arbeitet und auch rausfindet, zu Welchen Bundles diese Webseiten gehören. Damit kann man eine Möglichkeit haben, zu analysieren, welche Bundles belastet werden könnten.

Autor:

Yu

Definiert in Zeile 47 der Datei PageAccessTrackServlet.java.

Öffentliche Methoden

- void service (ServletRequest request, ServletResponse response) throws ServletException, IOException

Öffentliche, statische Methoden

- static List< String > getAppletsByBundleId (long bundleId)
- static List< String > getAppletsByUserId (String userId)
- static Map< String, String > getAppletUser ()
- static List< Bundle > getBundlesByUserId (String userId)
- static PageAccessTrackServlet getInstance ()
- static Map< String, String > getSessionLastAccessPage ()
- static Map< String, Timestamp > getSessionLastAccessTime ()

- static List<String> getUsersByBundleId (long bundleId)
- static void register (String alias, String applet, String user)
- static void removeUser (String userId)

Statische öffentliche Attribute

- static final String HOLD_SIGNAL = "hold"
- static final String REFRESH_SIGNAL = "refresh"
- static final String START_SIGNAL = "start"
- static final String STOP_SIGNAL = "stop"

1.10.11.2 Dokumentation der Elementfunktionen

static List<String> getAppletsByBundleId (long bundleId) [static]

Liefert die ID der Applets zurück, die gerade ein Bundle benutzen.

Parameter:

bundleId BundleId

Rückgabe:

eine Liste von Appletids

Definiert in Zeile 192 der Datei PageAccessTrackServlet.java.

static List<String> getAppletsByUserId (String userId) [static]

Liefert die IDs der Applets zurück, die ein Nutzer benutzt.

Parameter:

userId die Nutzerid

Rückgabe:

eine Liste von Appletids

Definiert in Zeile 213 der Datei PageAccessTrackServlet.java.

static Map<String, String> getAppletUser () [static]

Liefert das Map für Applet und Nutzer

Rückgabe:

Definiert in Zeile 232 der Datei PageAccessTrackServlet.java.

static List<Bundle> getBundlesByUserId (String userId) [static]

Liefert eine Liste von Bundleids zurück, die ein Nutzer gerade benutzt.

Parameter:

userId die ID des Nutzers

Rückgabe:

eine Liste von Ids der Bundles

Definiert in Zeile 243 der Datei PageAccessTrackServlet.java.

static PageAccessTrackServlet getInstance () [static]

Liefert die Instanz des Servlets zurück.

Rückgabe:

die Instanz des Servlets

Definiert in Zeile 265 der Datei PageAccessTrackServlet.java.

Benutzt PageAccessTrackServlet.cleanupTimer.

static Map<String, String> getSessionLastAccessPage () [static]

Liefert die letzten Zugriffsseiten der Sitzungen zurück

Rückgabe:

ein Map für Session-ID und URL

Definiert in Zeile 276 der Datei PageAccessTrackServlet.java.

static Map<String, Timestamp> getSessionLastAccessTime () [static]

Liefert die letzten Zugriffszeiten der Sitzungen zurück

Rückgabe:

ein Map für Session-ID und Zugriffszeit

Definiert in Zeile 285 der Datei PageAccessTrackServlet.java.

static List<String> getUsersByBundleId (long bundleId) [static]

Liefert die Liste von Nutzer-Id s zurück, die gerade ein Bundle benutzen.

Parameter:

bundleId Bundle-ID

Rückgabe:

eine Liste von Nutzer-IDs

Definiert in Zeile 296 der Datei PageAccessTrackServlet.java.

static void register (String alias, String applet, String user) [static]

Diese Methode findet zuerst heraus, in welchem Bundle liegt eine Webseite, dann notiert diese Information.

Parameter:

alias der URL einer Webseite

applet die Applet-ID

user die User-ID

Definiert in Zeile 322 der Datei PageAccessTrackServlet.java.

static void removeUser (String userId) [static]

Entfernt ein Nutzer, falls er sich aus dem System ausgeloggt oder er lange keine Aktion macht.

Parameter:

userId die Nutzer-ID

Definiert in Zeile 373 der Datei PageAccessTrackServlet.java.

void service (ServletRequest request, ServletResponse response) throws ServletException, IOException

Diese Methode handelt die Anfrage von Clients

Definiert in Zeile 440 der Datei PageAccessTrackServlet.java.

1.10.11.3 Dokumentation der Datenelemente**final String HOLD_SIGNAL = "hold" [static]**

Hold Signal: das Top-Menu-Applet lebt.

Definiert in Zeile 101 der Datei PageAccessTrackServlet.java.

final String REFRESH_SIGNAL = "refresh" [static]

Refresh Singal: die Webseite ist aktualisiert.

Definiert in Zeile 106 der Datei PageAccessTrackServlet.java.

final String START_SIGNAL = "start" [static]

Start Signal: clientseitig startet ein neue Top-Menu-Applet.

Definiert in Zeile 111 der Datei PageAccessTrackServlet.java.

final String STOP_SIGNAL = "stop" [static]

Stop Signal: das Top-Menu-Applet stoppt.

Definiert in Zeile 116 der Datei PageAccessTrackServlet.java.

1.10.12 SimpleMenuItem Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::api::MenuItem.

1.10.12.1 Ausführliche Beschreibung

Diese Klasse wird benutzt um einen einfachen Menüeintrag zu repräsentieren.

Autor:

Yu

Definiert in Zeile 11 der Datei SimpleMenuItem.java.

Öffentliche Methoden

- void accept (IMenuVisitor visitor)
- ItemType getType ()
- SimpleMenuItem (String id, String name, String url, String discription, String shortcut)
- SimpleMenuItem (String id, String name, String url, String discription)
- SimpleMenuItem (String id, String name, String url)
- SimpleMenuItem (String id, String name)
- SimpleMenuItem (IMenuItem another)

1.10.12.2 Beschreibung der Konstruktoren und Destruktoren

SimpleMenuItem (IMenuItem another)

Kopiekonstruktor

Parameter:

another ein anderer Menüeintrag

Definiert in Zeile 19 der Datei SimpleMenuItem.java.

SimpleMenuItem (String id, String name)

Konstruktor

Parameter:

id die ID des Menüeintrags

name der Name des Menüeintrags

Definiert in Zeile 31 der Datei SimpleMenuItem.java.

SimpleMenuItem (String id, String name, String url)

Konstruktor

Parameter:

id die ID des Menüeintrags

name der Name des Menüeintrags

url der Link des Menüeintrags

Definiert in Zeile 45 der Datei SimpleMenuItem.java.

SimpleMenuItem (String id, String name, String url, String discription)

Konstruktor

Parameter:

id die ID des Menüeintrags

name der Name des Menüeintrags

url der Link des Menüeintrags

discription die Beschreibung des Menüeintrags

Definiert in Zeile 61 der Datei SimpleMenuItem.java.

SimpleMenuItem (String id, String name, String url, String discription, String shortcut)

Konstruktor

Parameter:

id die ID des Menüeintrags

name der Name des Menüeintrags

url der Link des Menüeintrags

discription die Beschreibung des Menüeintrags

shortcut das Shortcut des Menüeintrags

Definiert in Zeile 79 der Datei SimpleMenuItem.java.

1.10.12.3 Dokumentation der Elementfunktionen

void accept (IMenuVisitor visitor)

Akzeptiert einen Menüvisitor

Implementiert IMenuItem.

Definiert in Zeile 88 der Datei SimpleMenuItem.java.

Benutzt IMenuVisitor.visit().

ItemType getType () [virtual]

Liefert den Typ des Menüeintrags zurück.

Implementiert IMenuItem.

Definiert in Zeile 96 der Datei SimpleMenuItem.java.

1.10.13 Submenu Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::api::MenuItem.

1.10.13.1 Ausführliche Beschreibung

Diese Klasse wird benutzt, um ein Submenü zu repräsentieren.

Autor:

Yu

Definiert in Zeile 16 der Datei Submenu.java.

Öffentliche Methoden

- void accept (IMenuVisitor visitor)
- final void addMenuItem (IMenuItem item)
- int generateMenuParams (int dynId, Map< String, String > newParams)
- final List< IMenuItem > getChildren ()
- final ItemType getType ()
- final void removeMenuItem (IMenuItem item)
- final void setChildren (List< IMenuItem > children)
- Submenu (String id, String name)
- Submenu (Submenu another)
- String toString ()

1.10.13.2 Beschreibung der Konstruktoren und Destruktoren

Submenu (Submenu another)

Kopiekonstruktor

Parameter:

another ein anderer Submenü

Definiert in Zeile 30 der Datei Submenu.java.

Benutzt Submenu.getChildren().

Submenu (String id, String name)

Konstruktor

Parameter:

id die ID des Menüeintrags

name der Name des Menüeintrags

Definiert in Zeile 43 der Datei Submenu.java.

Benutzt MenuItem.getParams().

1.10.13.3 Dokumentation der Elementfunktionen

void accept (IMenuVisitor visitor)

Akzeptiert einen Menüvisitor

Implementiert IMenuItem.

Definiert in Zeile 123 der Datei Submenu.java.

Benutzt IMenuVisitor.visit().

Wird benutzt von TopMenu.getTopMenuParams().

final void addMenuItem (IMenuItem item)

Fügt einen Menüeintrag hinzu.

Parameter:

item ein Menüeintrag

Definiert in Zeile 79 der Datei Submenu.java.

Benutzt Submenu.getChildren().

int generateMenuParams (int dynId, Map< String, String > newParams)

generiert die Menüparameter dieses Submenüs

Erneute Implementation von MenuItem.

Definiert in Zeile 105 der Datei Submenu.java.

Benutzt Submenu.getChildren().

final List<IMenuItem> getChildren () [virtual]

Liefert die Unterpunkte des Submenüs zurück.

Erneute Implementation von MenuItem.

Definiert in Zeile 60 der Datei Submenu.java.

Wird benutzt von AbstractAdoptableMenuContributor.addMenuContributor(), Submenu.addItem(), Submenu.generateMenuParams(), AbstractAdoptableMenuContributor.locateMenuItem(), AbstractAdoptableMenuContributor.removeMenuContributor(), Submenu.removeItem(), Submenu.Submenu() und JddmMenuParamsGenerator.visit().

final ItemType getType () [virtual]

Liefert den Typ des Submenüs zurück.

Implementiert IMenuItem.

Definiert in Zeile 52 der Datei Submenu.java.

final void removeMenuItem (IMenuItem item)

Entfernt einen Menüeintrag

Parameter:

item ein Menüeintrag

Definiert in Zeile 89 der Datei Submenu.java.

Benutzt Submenu.getChildren().

final void setChildren (List< IMenuItem > children)

Setter der Unterpunkte des Submenüs

Parameter:

children

Definiert in Zeile 69 der Datei Submenu.java.

String toString ()

als String

Erneute Implementation von MenuItem.

Definiert in Zeile 97 der Datei Submenu.java.

1.10.14 SysMsgManager Klassenreferenz**1.10.14.1 Ausführliche Beschreibung**

Diese Klasse kann von anderen Bundles benutzt, um eine Nachricht an eine Liste von Nutzern zu senden.

Autor:

Yu

Definiert in Zeile 23 der Datei SysMsgManager.java.

Öffentliche Methoden

- void notification (List< User > bundleUsers, String titel, String content)

Öffentliche, statische Methoden

- static SysMsgManager getInstance ()

1.10.14.2 Dokumentation der Elementfunktionen**static SysMsgManager getInstance () [static]**

Liefert die Instanz des SysMsgManagers zurück.

Rückgabe:

die SysMsgManager

Definiert in Zeile 35 der Datei SysMsgManager.java.

Benutzt SysMsgManager.instance.

void notification (List< User > bundleUsers, String titel, String content)

Diese Methode wird benutzt, eine Nachricht an eine Liste von Nutzern zu senden.

Parameter:

bundleUsers die Empfänger

titel der Titel der Nachricht

content der Inhalt der Nachricht

Definiert in Zeile 51 der Datei SysMsgManager.java.

Benutzt SysMessage.setPayload(), SysMessage.setReceiver() und SysEvent.toString().

1.10.15 CoreServer Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::impl::ITopMenuUpdateListener und de::offis::pg::eam::core::menu::impl::ModuleEventHandler.

1.10.15.1 Ausführliche Beschreibung

Diese Klasse wird dazu benutzt, um alle Socketverbindungen für Top-Menu-Applet von Clients zu behandeln. Wenn ein Modul installiert oder deinstalliert wird, muss häufig das Topmenu von Clients auch mit aktualisiert werden. In diesem Fall sorgt der CoreServer dafür, die erneute Menüparameter an Clients zu senden.

Autor:

Yu

Definiert in Zeile 32 der Datei CoreServer.java.

Öffentliche Methoden

- Set< Socket > getAllSockets ()
- Map< String, Socket > getAppletSocket ()
- ExecutorService getPool ()
- Map< String, Set< Socket > > getUserSockets ()
- Set< Socket > getUserSocketsByUserId (String userId)
- void handle (ModuleUninstalledEvent e)
- void handle (ModuleStoppedEvent e)
- void handle (ModuleStartedEvent e)
- void handle (ModuleInstalledEvent e)
- synchronized void handleMenuUpdate (Object evt, IMenuContributor menuContributor)
- void removeAllSockets ()
- void removeUserAllSockets (String userId)
- boolean removeUserSocket (String userId, Socket socket)
- void run ()
- void setUserSockets (Map< String, Set< Socket >> userSockets)

Öffentliche, statische Methoden

- `static CoreServer getInstance ()`

Statische öffentliche Attribute

- `static final String BEGIN_SIGNAL = "BEGIN"`
- `static final String HOLD_SIGNAL = "HOLD"`
- `static final int SYS_MSG_SERVER_MSG_TYPE = 1`
- `static final int TOP_MENU_SERVER_MSG_TYPE = 2`

1.10.15.2 Dokumentation der Elementfunktionen

Set<Socket> getAllSockets ()

Liefert alle Sockets zurück.

Rückgabe:

Definiert in Zeile 104 der Datei CoreServer.java.

Map<String, Socket> getAppletSocket ()

Liefert das Map für Applet und Socket zurück.

Rückgabe:

Definiert in Zeile 120 der Datei CoreServer.java.

static CoreServer getInstance () [static]

Liefert die CoreServer-Instanz zurück.

Rückgabe:

die CoreServer-Instanz

Definiert in Zeile 70 der Datei CoreServer.java.

Benutzt CoreServer.instance.

ExecutorService getPool ()

Liefert den Threadpool zurück.

Rückgabe:

Definiert in Zeile 129 der Datei CoreServer.java.

Map<String, Set<Socket> > getUserSockets ()

Liefert Nutzer's Sockets zurück.

Rückgabe:

Definiert in Zeile 138 der Datei CoreServer.java.

Set<Socket> getUserSocketsByUserId (String userId)

Liefer Sockets zurück, die zu einem Nutzer gehören.

Parameter:

userId die Nutzer-ID

Rückgabe:

eine Menge von Sockets

Definiert in Zeile 149 der Datei CoreServer.java.

Wird benutzt von CoreServer.removeUserAllSockets() und CoreServer.removeUserSocket().

void handle (ModuleUninstalledEvent e)

handelt das ModuleUninstalledEvent

Implementiert ModuleEventHandler.

Definiert in Zeile 178 der Datei CoreServer.java.

void handle (ModuleStoppedEvent e)

handlet das ModuleStoppedEvent

Implementiert ModuleEventHandler.

Definiert in Zeile 171 der Datei CoreServer.java.

void handle (ModuleStartedEvent e)

handelt das ModuleStartedEvent

Implementiert ModuleEventHandler.

Definiert in Zeile 164 der Datei CoreServer.java.

void handle (ModuleInstalledEvent e)

handelt das ModuleInstalledEvent

Implementiert ModuleEventHandler.

Definiert in Zeile 157 der Datei CoreServer.java.

synchronized void handleMenuUpdate (Object evt, IMenuContributor menuContributor)

handlet die Aktualisierung des Topmenu

Implementiert ITopMenuUpdateListener.

Definiert in Zeile 185 der Datei CoreServer.java.

void removeAllSockets ()

Entfernt alle Sockets.

Definiert in Zeile 197 der Datei CoreServer.java.

void removeUserAllSockets (String userId)

Entfernt alle Sockets, die zu einem Nutzer gehören.

Parameter:

userId

Definiert in Zeile 206 der Datei CoreServer.java.

Benutzt CoreServer.getUserSocketsByUserId().

boolean removeUserSocket (String userId, Socket socket)

Entfernt ein Socket eines Nutzers

Parameter:

userId die Nutzerid

socket das Socket

Rückgabe:

Definiert in Zeile 222 der Datei CoreServer.java.

Benutzt CoreServer.getUserSocketsByUserId().

void run ()

Diese Methode wird aufgerufen, wenn CoreServer gestartet wird. Hier lauscht der CoreServer an neue Socketverbindung von Clients.

Definiert in Zeile 235 der Datei CoreServer.java.

void setUserSockets (Map< String, Set< Socket >> userSockets)

Setter des Map für Nutzer und Socket

Parameter:

userSockets das neue Map für Nutzer und Sockets

Definiert in Zeile 334 der Datei CoreServer.java.

1.10.15.3 Dokumentation der Datenelemente

final String BEGIN_SIGNAL = "BEGIN" [static]

Start Singal für die Socketverbindung

Definiert in Zeile 38 der Datei CoreServer.java.

final String HOLD_SIGNAL = "HOLD" [static]

Hold Singal für die Socketverbindung

Definiert in Zeile 43 der Datei CoreServer.java.

final int SYS_MSG_SERVER_MSG_TYPE = 1 [static]

Konstante für den Typ der Systemnachrichten

Definiert in Zeile 48 der Datei CoreServer.java.

final int TOP_MENU_SERVER_MSG_TYPE = 2 [static]

Konstante für den Typ der Topmenunachrichten

Definiert in Zeile 53 der Datei CoreServer.java.

1.10.16 DefaultMenuVisitor Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::impl::IMenuVisitor.

Basisklasse für JddmMenuParamsGenerator.

1.10.16.1 Ausführliche Beschreibung

Diese Klasse ist eine defaulte Implementierung der Schnittstelle `IMenuVisitor`.

Autor:

Yu

Definiert in Zeile 14 der Datei `DefaultMenuVisitor.java`.

Öffentliche Methoden

- `void visit (AuthMenuItem item)`
- `void visit (Submenu item)`
- `void visit (SimpleMenuItem item)`

1.10.16.2 Dokumentation der Elementfunktionen

void visit (AuthMenuItem item)

Besucht ein `AuthMenuItem`

Implementiert `IMenuVisitor`.

Erneute Implementation in `JddmMenuParamsGenerator`.

Definiert in Zeile 34 der Datei `DefaultMenuVisitor.java`.

void visit (Submenu item)

Besucht ein `Submenu`

Implementiert `IMenuVisitor`.

Erneute Implementation in `JddmMenuParamsGenerator`.

Definiert in Zeile 27 der Datei `DefaultMenuVisitor.java`.

void visit (SimpleMenuItem item)

Besucht ein `SimpleMenuItem`

Implementiert `IMenuVisitor`.

Erneute Implementation in `JddmMenuParamsGenerator`.

Definiert in Zeile 20 der Datei `DefaultMenuVisitor.java`.

1.10.17 FacesServletWrapper Klassenreferenz

Abgeleitet von `javax::servlet::Servlet`.

1.10.17.1 Ausführliche Beschreibung

Diese Klasse ist ein Wrapper der Klasse FacesServlet. Die Instanz dieser Klasse wird statt der Instanz der Klasse FacesServlet in dem HttpService registriert. D.h. wenn eine Anfrag ankommt, wird die zuerst durch die FacesServletWrapper bearbeitet und dann an die FacesServlet weitergeleitet. Dieser Umweg hier ermöglicht z.B. die Registerierung der letzten Zugriffszeit einer Session.

Autor:

Yu

Definiert in Zeile 29 der Datei FacesServletWrapper.java.

Öffentliche Methoden

- void destroy ()
- ServletConfig getServletConfig ()
- String getServletInfo ()
- void init (ServletConfig config) throws ServletException
- void service (ServletRequest request, ServletResponse response) throws ServletException, IOException

1.10.17.2 Dokumentation der Elementfunktionen

void destroy ()

Diese Methode wird aufgerufen, wenn ein Servlet vernichtet wird.

Definiert in Zeile 39 der Datei FacesServletWrapper.java.

ServletConfig getServletConfig ()

Liefert die Servletkonfiguration zurück.

Definiert in Zeile 46 der Datei FacesServletWrapper.java.

String getServletInfo ()

Liefert die Servletinformation zurück.

Definiert in Zeile 53 der Datei FacesServletWrapper.java.

void init (ServletConfig config) throws ServletException

Initialisiert ein Servlet.

Definiert in Zeile 60 der Datei FacesServletWrapper.java.

void service (ServletRequest request, ServletResponse response) throws ServletException, IOException

Bearbeitet die Anfragen aus Clients.

Definiert in Zeile 67 der Datei FacesServletWrapper.java.

1.10.18 IAccessRight Schnittstellenreferenz

Basisklasse für AuthMenuItem.

1.10.18.1 Ausführliche Beschreibung

Diese Schnittstelle definiert die Methoden, die ein zu autorisierendes Menü implementieren muss.

Autor:

Yu

Definiert in Zeile 10 der Datei IAccessRight.java.

Öffentliche Methoden

- String getView ()
- void setView (String view)

1.10.18.2 Dokumentation der Elementfunktionen

String getView ()

Liefert das View zurück

Rückgabe:

der Name eines View

Implementiert in AuthMenuItem.

void setView (String view)

Setter des Namens eines View

Parameter:

view der Name eines View

Implementiert in AuthMenuItem.

1.10.19 IMenuVisitor Schnittstellenreferenz

Basisklasse für DefaultMenuVisitor.

1.10.19.1 Ausführliche Beschreibung

Zur Generierung der Menüparameter wird der Entwurfsmuster Visitor benutzt. Diese Schnittstelle definiert alle Methoden, die ein Visitor implementieren muss.

Autor:

Yu

Definiert in Zeile 15 der Datei IMenuVisitor.java.

Öffentliche Methoden

- void visit (AuthMenuItem item)
- void visit (Submenu item)
- void visit (SimpleMenuItem item)

1.10.19.2 Dokumentation der Elementfunktionen

void visit (AuthMenuItem item)

Besucht ein AuthMenuItem

Parameter:

item ein AuthMenuItem

Implementiert in DefaultMenuVisitor und JddmMenuParamsGenerator.

void visit (Submenu item)

Besucht ein Submenu

Parameter:

item ein Submenu

Implementiert in DefaultMenuVisitor und JddmMenuParamsGenerator.

void visit (SimpleMenuItem item)

Besucht ein SimpleMenuItem

Parameter:

item ein SimpleMenuItem

Implementiert in `DefaultMenuVisitor` und `JddmMenuParamsGenerator`.

Wird benutzt von `Submenu.accept()`, `SimpleMenuItem.accept()` und `AuthMenuItem.accept()`.

1.10.20 ITopMenuUpdateListener Schnittstellenreferenz

Basisklasse für `CoreServer`.

1.10.20.1 Ausführliche Beschreibung

Wenn eine Klasse Interesse an der Aktualisierung des Topmenu hat, soll diese Schnittstelle implementieren.

Autor:

Yu

Definiert in Zeile 12 der Datei `ITopMenuUpdateListener.java`.

Öffentliche Methoden

- `void handleMenuUpdate (Object evt, IMenuContributor menuContributor)`

1.10.20.2 Dokumentation der Elementfunktionen

`void handleMenuUpdate (Object evt, IMenuContributor menuContributor)`

Wenn das Topmenu aktualisiert, dann wird diese Methode aufgerufen.

Parameter:

evt das Ereignis der Aktualisierung

menuContributor das neue hinzugefügte oder entfernte `MenuContributor`

Implementiert in `CoreServer`.

1.10.21 JddmMenuParamsGenerator Klassenreferenz

Abgeleitet von `de::offis::pgg::eam::core::menu::impl::DefaultMenuVisitor`.

1.10.21.1 Ausführliche Beschreibung

Diese Klasse wird zur Generierung der Menüparameter benutzt und implementiert alle Methoden der Schnittstelle `IMenuVisitor`. Die generierten Parameter werden mit dem JDDM-Applet anpassen, damit das JDDM-Applet das Menü richtig darstellen kann. Dabei wird aufgepasst, dass Menüeinträge mit Views verknüpfen könnten. Für solche

Menüeinträge werden die Rechte und zwar Views überprüft, bevor ihre Menüparameter generiert werden. Das sieht so aus, dass manche Menüeinträge ausgeblendet werden, wenn ein Nutzer kein Recht dafür hat.

Autor:

Yu

Definiert in Zeile 30 der Datei JddmMenuParamsGenerator.java.

Öffentliche Methoden

- String getMenuParams ()
- JddmMenuParamsGenerator (String target, Set< Pair > views)
- Map< String, String > resolveParams (String params, String split)
- void visit (Submenu submenu)
- void visit (SimpleMenuItem simpleMenuItem)
- void visit (AuthMenuItem authorizableMenuItem)

1.10.21.2 Beschreibung der Konstruktoren und Destruktoren

JddmMenuParamsGenerator (String target, Set< Pair > views)

Konstruktor

Parameter:

target wo ein Menü aufgehängt werden soll.

views views und zwar Rechte

Definiert in Zeile 60 der Datei JddmMenuParamsGenerator.java.

1.10.21.3 Dokumentation der Elementfunktionen

String getMenuParams ()

Liefert die generierten Menüparameter zurück.

Rückgabe:

Menüparameter als String

Definiert in Zeile 70 der Datei JddmMenuParamsGenerator.java.

Benutzt PrintfFormat.printf().

Wird benutzt von TopMenu.getTopMenuParams().

Map<String, String> resolveParams (String params, String split)

Splittet Menüparameter.

Parameter:

params Menüparameter in String getrennt durch “;”

split Zeichen für die Trennung, z.b. “;”

Rückgabe:

die gesplitteten Parameter

Definiert in Zeile 94 der Datei JddmMenuParamsGenerator.java.

void visit (Submenu submenu)

Besucht ein Submenu

Erneute Implementation von DefaultMenuVisitor.

Definiert in Zeile 176 der Datei JddmMenuParamsGenerator.java.

Benutzt Submenu.getChildren(), MenuItem.getLocaleName(), MenuItem.getParam() und MenuItem.getParams().

void visit (SimpleMenuItem simpleMenuItem)

Besucht ein SimpleMenuItem

Erneute Implementation von DefaultMenuVisitor.

Definiert in Zeile 153 der Datei JddmMenuParamsGenerator.java.

Benutzt MenuItem.getLocaleName() und MenuItem.getParams().

void visit (AuthMenuItem authorizableMenuItem)

Besucht ein AuthMenuItem.

Erneute Implementation von DefaultMenuVisitor.

Definiert in Zeile 123 der Datei JddmMenuParamsGenerator.java.

Benutzt IMenuItem.accept(), AuthMenuItem.getBundleName(), AuthMenuItem.getDelegate(), IMenuItem.getId() und AuthMenuItem.getView().

1.10.22 ModuleEvent Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::impl::SysEvent.

Basisklasse für ModuleInstalledEvent, ModuleStartedEvent, ModuleStoppedEvent und ModuleUninstalledEvent.

1.10.22.1 Ausführliche Beschreibung

Diese Klasse packt die low-level OSGI-Ereignisse, wie z.B. BundleEvent.INSTALLED, BundleEvent.STARTED, BundleEvent.STOPPED und BundleEvent.UNINSTALLED.

Autor:

Yu

Definiert in Zeile 13 der Datei ModuleEvent.java.

Öffentliche Methoden

- final String getAction ()
- final long getBundleId ()
- final String getBundleName ()
- ModuleEvent (BundleEvent event)
- final void setAction (String action)
- final void setBundleId (long bundleId)
- final void setBundleName (String bundleName)

Statische öffentliche Attribute

- static final String ACTION = "ACTION"
- static final String BUNDLE_ID = "BUNDLE_ID"
- static final String BUNDLE_NAME = "BUNDLE_NAME"
- static final String MODULE_INSTALLED_EVENT = "MODULE_INSTALLED_EVENT"
- static final String MODULE_STARTED_EVENT = "MODULE_STARTED_EVENT"
- static final String MODULE_UNINSTALLED_EVENT = "MODULE_UNINSTALLED_EVENT"
- static final String MOUDLE_STOPPED_EVENT = "MOUDLE_STOPPED_EVENT"
- static final String SHUTDOWN_EVENT = "SHUTDOWN_EVENT"

1.10.22.2 Beschreibung der Konstruktoren und Destruktoren

ModuleEvent (BundleEvent event)

Konstruktor

Parameter:

event das Bundleereignis

Definiert in Zeile 61 der Datei ModuleEvent.java.

Benutzt SysEvent.put(), ModuleEvent.setBundleId(), ModuleEvent.setBundleName() und SysEvent.setType().

1.10.22.3 Dokumentation der Elementfunktionen

final String getAction ()

Liefert die Aktion zurück.

Rückgabe:

die Aktion

Definiert in Zeile 111 der Datei ModuleEvent.java.

final long getBundleId ()

Liefert Bundle-ID zurück.

Rückgabe:

die Bundle-ID

Definiert in Zeile 73 der Datei ModuleEvent.java.

final String getBundleName ()

Liefert den Bundlenamen zurück.

Rückgabe:

der Bundlename

Definiert in Zeile 92 der Datei ModuleEvent.java.

final void setAction (String action)

Setter der Aktion

Parameter:

die Aktion

Definiert in Zeile 121 der Datei ModuleEvent.java.

Benutzt SysEvent.put().

final void setBundleId (long bundleId)

Setter der Bundle-ID

Parameter:

bundleId die neue Bundle-ID

Definiert in Zeile 83 der Datei ModuleEvent.java.

Benutzt SysEvent.put().

Wird benutzt von ModuleEvent.ModuleEvent().

final void setBundleName (String bundleName)

Setter des Bundlenamens

Parameter:

bundleName der neue Bundlename

Definiert in Zeile 102 der Datei ModuleEvent.java.

Benutzt SysEvent.put().

Wird benutzt von ModuleEvent.ModuleEvent().

1.10.22.4 Dokumentation der Datenelemente

final String ACTION = "ACTION" [static]

Konstante für Aktion

Definiert in Zeile 53 der Datei ModuleEvent.java.

final String BUNDLE_ID = "BUNDLE_ID" [static]

Konstante für Bundle-ID

Definiert in Zeile 43 der Datei ModuleEvent.java.

final String BUNDLE_NAME = "BUNDLE_NAME" [static]

Konstante für Bundlenamen

Definiert in Zeile 48 der Datei ModuleEvent.java.

final String MODULE_INSTALLED_EVENT = "MODULE_INSTALLED_EVENT" [static]

Ereignis für die Installtion eines Moduls

Definiert in Zeile 18 der Datei ModuleEvent.java.

final String MODULE_STARTED_EVENT = "MODULE_STARTED_EVENT" [static]

Ereignis für den Start eines Moduls

Definiert in Zeile 23 der Datei ModuleEvent.java.

```
final String MODULE_UNINSTALLED_EVENT =  
"MODULE_UNINSTALLED_EVENT" [static]
```

Ereignis für die Deinstalltion eines Moduls

Definiert in Zeile 33 der Datei ModuleEvent.java.

```
final String MOUDLE_STOPPED_EVENT = "MOUDLE_STOPPED_EVENT"  
[static]
```

Ereignis für die Installtion eines Moduls

Definiert in Zeile 28 der Datei ModuleEvent.java.

```
final String SHUTDOWN_EVENT = "SHUTDOWN_EVENT" [static]
```

Ereignis für das Hunterfahren des EAM-Systems

Definiert in Zeile 38 der Datei ModuleEvent.java.

1.10.23 ModuleEventDispatcher Klassenreferenz

Abgeleitet von org::osgi::framework::BundleListener.

1.10.23.1 Ausführliche Beschreibung

Diese Klasse transformiert ein OSGI-Ereignis zu einem EAM-Ereignis

Autor:

Yu

Definiert in Zeile 12 der Datei ModuleEventDispatcher.java.

Öffentliche Methoden

- void bundleChanged (BundleEvent event)
- ModuleEventDispatcher (ModuleEventHandler handler)

Paketattribute

- ModuleEventHandler handler

1.10.23.2 Beschreibung der Konstruktoren und Destruktoren

ModuleEventDispatcher (ModuleEventHandler handler)

Konstruktor

Parameter:

handler ein Ereignishandler

Definiert in Zeile 23 der Datei ModuleEventDispatcher.java.

Benutzt ModuleEventDispatcher.handler.

1.10.23.3 Dokumentation der Elementfunktionen

void bundleChanged (BundleEvent event)

handelt OSGI-Bundle-Event

Definiert in Zeile 31 der Datei ModuleEventDispatcher.java.

Benutzt ModuleEventHandler.handle() und ModuleEventDispatcher.handler.

1.10.23.4 Dokumentation der Datenelemente

ModuleEventHandler handler [package]

Ereignishandler

Definiert in Zeile 17 der Datei ModuleEventDispatcher.java.

Wird benutzt von ModuleEventDispatcher.bundleChanged() und ModuleEventDispatcher.ModuleEventDispatcher().

1.10.24 ModuleEventHandler Schnittstellenreferenz

Basisklasse für CoreServer.

1.10.24.1 Ausführliche Beschreibung

Diese Schnittstelle definiert die Methoden, die ein ModuleEventHandler implementieren muss, falls eine Klasse Interesse an Moduleereignissen hat.

Autor:

Yu

Definiert in Zeile 10 der Datei ModuleEventHandler.java.

Öffentliche Methoden

- void handle (ModuleUninstalledEvent e)
- void handle (ModuleStoppedEvent e)
- void handle (ModuleStartedEvent e)
- void handle (ModuleInstalledEvent e)

1.10.24.2 Dokumentation der Elementfunktionen

void handle (ModuleUninstalledEvent e)

handelt ModuleUninstalledEvent

Parameter:

e ein ModuleUninstalledEvent

Implementiert in CoreServer.

void handle (ModuleStoppedEvent e)

handelt ModuleStoppedEvent

Parameter:

e ein ModuleStoppedEvent

Implementiert in CoreServer.

void handle (ModuleStartedEvent e)

handelt ModuleStartedEvent

Parameter:

e ein ModuleStartedEvent

Implementiert in CoreServer.

void handle (ModuleInstalledEvent e)

handelt ModuleInstalledEvent

Parameter:

e ein ModuleInstalledEvent

Implementiert in CoreServer.

Wird benutzt von ModuleEventDispatcher.bundleChanged().

1.10.25 ModuleInstalledEvent Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::impl::ModuleEvent.

1.10.25.1 Ausführliche Beschreibung

Diese Klasse kapselt ein Bundle-Install-Event.

Autor:

Yu

Definiert in Zeile 11 der Datei `ModuleInstalledEvent.java`.

1.10.26 ModuleStartedEvent Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::menu::impl::ModuleEvent`.

1.10.26.1 Ausführliche Beschreibung

Diese Klasse kapselt ein Bundle-Started-Event

Autor:

Yu

Definiert in Zeile 10 der Datei `ModuleStartedEvent.java`.

1.10.27 ModuleStoppedEvent Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::menu::impl::ModuleEvent`.

1.10.27.1 Ausführliche Beschreibung

Diese Klasse kapselt ein Bundle-Stopped-Event.

Autor:

Yu

Definiert in Zeile 10 der Datei `ModuleStoppedEvent.java`.

1.10.28 ModuleUninstalledEvent Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::menu::impl::ModuleEvent`.

1.10.28.1 Ausführliche Beschreibung

Diese Klasse kapselt ein Bundle-Uninstalled-Event.

Autor:

Yu

Definiert in Zeile 10 der Datei `ModuleUninstalledEvent.java`.

1.10.29 Notification Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::menu::impl::SysEvent`.

1.10.29.1 Ausführliche Beschreibung

Diese Klasse kapselt eine Systemnachricht.

Autor:

Yu

Definiert in Zeile 9 der Datei `Notification.java`.

1.10.30 PrintfFormat Klassenreferenz

1.10.30.1 Ausführliche Beschreibung

`PrintfFormat` allows the formatting of an array of objects embedded within a string. Primitive types must be passed using wrapper types. The formatting is controlled by a control string.

A control string is a Java string that contains a control specification. The control specification starts at the first percent sign (%) in the string, provided that this percent sign

1. is not escaped protected by a matching % or is not an escape % character,
2. is not at the end of the format string, and
3. precedes a sequence of characters that parses as a valid control specification.

A control specification usually takes the form:

```
% ['-+ #0]* [0..9]* { . [0..9]* }+  
    { [h1L] }+ [idfgGoxXeEcs]
```

There are variants of this basic form that are discussed below.

The format is composed of zero or more directives defined as follows:

- ordinary characters, which are simply copied to the output stream;
- escape sequences, which represent non-graphic characters; and
- conversion specifications, each of which results in the fetching of zero or more arguments.

The results are undefined if there are insufficient arguments for the format. Usually an unchecked exception will be thrown. If the format is exhausted while arguments remain,

the excess arguments are evaluated but are otherwise ignored. In format strings containing the % form of conversion specifications, each argument in the argument list is used exactly once.

Conversions can be applied to the *n*th argument after the format in the argument list, rather than to the next unused argument. In this case, the conversion character % is replaced by the sequence %*n*\$, where *n* is a decimal integer giving the position of the argument in the argument list.

In format strings containing the %*n*\$ form of conversion specifications, each argument in the argument list is used exactly once.

Escape Sequences The following table lists escape sequences and associated actions on display devices capable of the action.

Sequence	Name	Description
\	backlash	None.
	alert	Attempts to alert the user through audible or visible notification.
	backspace	Moves the printing position to one column before the current position, unless the current position is the start of a line.
	form-feed	Moves the printing position to the initial printing position of the next logical page.
	newline	Moves the printing position to the start of the next line.
	carriage-return	Moves the printing position to the start of the current line.
	tab	Moves the printing position to the next implementation- defined horizontal tab position.
	vertical-tab	Moves the printing position to the start of the next implementation-defined vertical tab position.

Conversion Specifications Each conversion specification is introduced by the percent sign character (%). After the character %, the following appear in sequence:

Zero or more flags (in any order), which modify the meaning of the conversion

specification.

An optional minimum field width. If the converted value has fewer characters than the field width, it will be padded with spaces by default on the left; t will be padded on the right, if the left- adjustment flag (-), described below, is given to the field width. The field width takes the form of a decimal integer. If the conversion character is s, the field width is the the minimum number of characters to be printed.

An optional precision that gives the minumum number of digits to appear for the d, i, o, x or X conversions (the field is padded with leading zeros); the number of digits to appear after the radix character for the e, E, and f conversions, the maximum number of significant digits for the g and G conversions; or the maximum number of characters to be written from a string is s and S conversions. The precision takes the form of an optional decimal digit string, where a null digit string is treated as 0. If a precision appears with a c conversion character the precision is ignored.

An optional h specifies that a following d, i, o, x, or X conversion character applies to a type short argument (the argument will be promoted according to the integral promotions and its value converted to type short before printing).

An optional l (ell) specifies that a following d, i, o, x, or X conversion character applies to a type long argument.

A field width or precision may be indicated by an asterisk (*) instead of a digit string. In this case, an integer argument supplised the field width precision. The argument that is actually converted is not fetched until the conversion letter is seen, so the the arguments specifying field width or precision must appear before the argument (if any) to be converted. If the precision argument is negative, it will be changed to zero. A negative field width argument is taken as a - flag, followed by a positive field width.

In format strings containing the %n\$ form of a conversion specification, a field width or precision may be indicated by the sequence *m\$, where m is a decimal integer giving the position in the argument list (after the format argument) of an integer argument containing the field width or precision.

The format can contain either numbered argument specifications (that is, %n\$ and *m\$), or unnumbered argument specifications (that is % and *), but normally not both. The only exception to this is that %% can be mixed with the %n\$ form. The results of mixing numbered and unnumbered argument specifications in a format string are undefined.

Flag Characters The flags and their meanings are:

- ' integer portion of the result of a decimal conversion (i, d, f, g, or G) will be formatted with thousands' grouping characters. For other conversions the flag is ignored. The non-monetary grouping character is used.
- result of the conversion is left-justified within the field. (It will be right-justified if this flag is not specified).
- + result of a signed conversion always begins with a sign (+ or -). (It will begin with a sign only when a negative value is converted if this flag is not specified.)

- <space>** If the first character of a signed conversion is not a sign, a space character will be placed before the result. This means that if the space character and + flags both appear, the space flag will be ignored.
- #** value is to be converted to an alternative form. For c, d, i, and s conversions, the flag has no effect. For o conversion, it increases the precision to force the first digit of the result to be a zero. For x or X conversion, a non-zero result has 0x or 0X prefixed to it, respectively. For e, E, f, g, and G conversions, the result always contains a radix character, even if no digits follow the radix character (normally, a decimal point appears in the result of these conversions only if a digit follows it). For g and G conversions, trailing zeros will not be removed from the result as they normally are.
- 0** d, i, o, x, X, e, E, f, g, and G conversions, leading zeros (following any indication of sign or base) are used to pad to the field width; no space padding is performed. If the 0 and - flags both appear, the 0 flag is ignored. For d, i, o, x, and X conversions, if a precision is specified, the 0 flag will be ignored. For c conversions, the flag is ignored.

Conversion Characters Each conversion character results in fetching zero or more arguments. The results are undefined if there are insufficient arguments for the format. Usually, an unchecked exception will be thrown. If the format is exhausted while arguments remain, the excess arguments are ignored.

The conversion characters and their meanings are:

- d,i** The int argument is converted to a signed decimal in the style [-]dddd. The precision specifies the minimum number of digits to appear; if the value being converted can be represented in fewer digits, it will be expanded with leading zeros. The default precision is 1. The result of converting 0 with an explicit precision of 0 is no characters.
- o** The int argument is converted to unsigned octal format in the style ddddd. The precision specifies the minimum number of digits to appear; if the value being converted can be represented in fewer digits, it will be expanded with leading zeros. The default precision is 1. The result of converting 0 with an explicit precision of 0 is no characters.
- x** The int argument is converted to unsigned hexadecimal format in the style dddd; the letters abcdef are used. The precision specifies the minimum number of digits to appear; if the value being converted can be represented in fewer digits, it will be expanded with leading zeros. The default precision is 1. The result of converting 0 with an explicit precision of 0 is no characters.
- X** Behaves the same as the x conversion character except that letters ABCDEF are used instead of abcdef.
- f** The floating point number argument is written in decimal notation in the style [-]ddd.ddd, where the number of digits after the radix character (shown here as a decimal point) is equal to the precision specification. A Locale is used to determine the radix character to use in this format. If the precision is omitted from the argument, six digits are written after the radix character; if the precision is explicitly

0 and the # flag is not specified, no radix character appears. If a radix character appears, at least 1 digit appears before it. The value is rounded to the appropriate number of digits.

- e,E** The floating point number argument is written in the style [-]d.ddde{+-}dd (the symbols {+-} indicate either a plus or minus sign), where there is one digit before the radix character (shown here as a decimal point) and the number of digits after it is equal to the precision. A Locale is used to determine the radix character to use in this format. When the precision is missing, six digits are written after the radix character; if the precision is 0 and the # flag is not specified, no radix character appears. The E conversion will produce a number with E instead of e introducing the exponent. The exponent always contains at least two digits. However, if the value to be written requires an exponent greater than two digits, additional exponent digits are written as necessary. The value is rounded to the appropriate number of digits.
- g,G** The floating point number argument is written in style f or e (or in style E in the case of a G conversion character), with the precision specifying the number of significant digits. If the precision is zero, it is taken as one. The style used depends on the value converted: style e (or E) will be used only if the exponent resulting from the conversion is less than -4 or greater than or equal to the precision. Trailing zeros are removed from the result. A radix character appears only if it is followed by a digit.
- c,C** The integer argument is converted to a char and the result is written.
- s,S** The argument is taken to be a string and bytes from the string are written until the end of the string or the number of bytes indicated by the precision specification of the argument is reached. If the precision is omitted from the argument, it is taken to be infinite, so all characters up to the end of the string are written.

% Write a % character; no argument is converted.

If a conversion specification does not match one of the above forms, an `IllegalArgumentException` is thrown and the instance of `PrintfFormat` is not created.

If a floating point value is the internal representation for infinity, the output is `[+]Infinity`, where `Infinity` is either `Infinity` or `Inf`, depending on the desired output string length. Printing of the sign follows the rules described above.

If a floating point value is the internal representation for "not-a-number," the output is `[+]NaN`. Printing of the sign follows the rules described above.

In no case does a non-existent or small field width cause truncation of a field; if the result of a conversion is wider than the field width, the field is simply expanded to contain the conversion result.

The behavior is like `printf`. One exception is that the minimum number of exponent digits is 3 instead of 2 for e and E formats when the optional L is used before the e, E, g, or G conversion character. The optional L does not imply conversion to a long long double.

The biggest divergence from the C `printf` specification is in the use of 16 bit characters. This allows the handling of characters beyond the small ASCII character set and allows the utility to interoperate correctly with the rest of the Java runtime environment.

Omissions from the C printf specification are numerous. All the known omissions are present because Java never uses bytes to represent characters and does not have pointers:

- c is the same as C.
- s is the same as S.
- u, p, and n conversion characters.
- ws format.
- h modifier applied to an n conversion character.
- l (ell) modifier applied to the c, n, or s conversion characters.
- ll (ell ell) modifier to d, i, o, u, x, or X conversion characters.
- ll (ell ell) modifier to an n conversion character.
- c, C, d,i,o,u,x, and X conversion characters apply to Byte, Character, Short, Integer, Long types.
- f, e, E, g, and G conversion characters apply to Float and Double types.
- s and S conversion characters apply to String types.
- All other reference types can be formatted using the s or S conversion characters only.

Most of this specification is quoted from the Unix man page for the sprintf utility.

Autor:

Allan Jacobs

Version:

1 Release 1: Initial release. Release 2: Asterisk field widths and precisions n\$ and *m\$
Bug fixes g format fix (2 digits in e form corrupt) rounding in f format implemented
round up when digit not printed is 5 formatting of -0.0f round up/down when last
digits are 50000...

Definiert in Zeile 444 der Datei PrintfFormat.java.

Öffentliche Methoden

- PrintfFormat (Locale locale, String fmtArg) throws IllegalArgumentException
- PrintfFormat (String fmtArg) throws IllegalArgumentException
- String sprintf (Object x) throws IllegalArgumentException
- String sprintf (String x) throws IllegalArgumentException
- String sprintf (double x) throws IllegalArgumentException
- String sprintf (long x) throws IllegalArgumentException
- String sprintf (int x) throws IllegalArgumentException
- String sprintf ()

- `String sprintf (Object[] o)`

1.10.30.2 Beschreibung der Konstruktoren und Destruktoren

PrintfFormat (String fmtArg) throws IllegalArgumentException

Constructs an array of control specifications possibly preceded, separated, or followed by ordinary strings. Control strings begin with unpaired percent signs. A pair of successive percent signs designates a single percent sign in the format.

Parameter:

fmtArg Control string.

Ausnahmebehandlung:

IllegalArgumentException if the control string is null, zero length, or otherwise malformed.

Definiert in Zeile 457 der Datei `PrintfFormat.java`.

PrintfFormat (Locale locale, String fmtArg) throws IllegalArgumentException

Constructs an array of control specifications possibly preceded, separated, or followed by ordinary strings. Control strings begin with unpaired percent signs. A pair of successive percent signs designates a single percent sign in the format.

Parameter:

fmtArg Control string.

Ausnahmebehandlung:

IllegalArgumentException if the control string is null, zero length, or otherwise malformed.

Definiert in Zeile 473 der Datei `PrintfFormat.java`.

1.10.30.3 Dokumentation der Elementfunktionen

String sprintf (Object x) throws IllegalArgumentException

Format an Object. Convert wrapper types to their primitive equivalents and call the appropriate internal formatting method. Convert Strings using an internal formatting method for Strings. Otherwise use the default formatter (use `toString`).

Parameter:

x the Object to format.

Rückgabe:

the formatted String.

Ausnahmebehandlung:

IllegalArgumentException if the conversion character is inappropriate for formatting an unwrapped value.

Definiert in Zeile 735 der Datei PrintfFormat.java.

String sprintf (String x) throws IllegalArgumentException

Format a String.

Parameter:

x The String to format.

Rückgabe:

The formatted String.

Ausnahmebehandlung:

IllegalArgumentException if the conversion character is neither s nor S.

Definiert in Zeile 706 der Datei PrintfFormat.java.

String sprintf (double x) throws IllegalArgumentException

Format a double.

Parameter:

x The double to format.

Rückgabe:

The formatted String.

Ausnahmebehandlung:

IllegalArgumentException if the conversion character is c, C, s, S, d, D, x, X, or o.

Definiert in Zeile 683 der Datei PrintfFormat.java.

String sprintf (long x) throws IllegalArgumentException

Format an long.

Parameter:

x The long to format.

Rückgabe:

The formatted String.

Ausnahmebehandlung:

IllegalArgumentException if the conversion character is f, e, E, g, G, s, or S.

Definiert in Zeile 659 der Datei PrintfFormat.java.

String sprintf (int x) throws IllegalArgumentException

Format an int.

Parameter:

- x** The int to format.

Rückgabe:

The formatted String.

Ausnahmebehandlung:

IllegalArgumentException if the conversion character is f, e, E, g, G, s, or S.

Definiert in Zeile 635 der Datei PrintfFormat.java.

String sprintf ()

Format nothing. Just use the control string.

Rückgabe:

the formatted String.

Definiert in Zeile 613 der Datei PrintfFormat.java.

String sprintf (Object[] o)

Format an array of objects. Byte, Short, Integer, Long, Float, Double, and Character arguments are treated as wrappers for primitive types.

Parameter:

- o** The array of objects to format.

Rückgabe:

The formatted String.

Definiert in Zeile 542 der Datei PrintfFormat.java.

Wird benutzt von JddmMenuParamsGenerator.getMenuParams(), SysMessage.toString() und SysEvent.toString().

1.10.31 SessionClosed Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::impl::SysEvent.

1.10.31.1 Ausführliche Beschreibung

Diese Klasse wird benutzt, wenn ein Session abläuft.

Autor:

Yu

Definiert in Zeile 9 der Datei SessionClosed.java.

1.10.32 SessionManager Klassenreferenz

Abgeleitet von javax:servlet:http::HttpSessionBindingListener.

1.10.32.1 Ausführliche Beschreibung

Diese Klasse wird dazu benutzt, die Sitzung im EAM-System zu verwalten. Wenn sich ein Nutzer einloggt oder ausloggt, wird der **SessionManager** seinen Register aktualisieren. Wenn sich ein Nutzer auf einer Maschine eingeloggt hat und er sich an einer anderen Maschine wieder erneut einloggt, dann wird die alte Sitzung automatisch gebrochen wird.

Autor:

Yu

Definiert in Zeile 24 der Datei SessionManager.java.

Öffentliche Methoden

- void valueBound (HttpSessionBindingEvent event)
- void valueUnbound (HttpSessionBindingEvent event)

Öffentliche, statische Methoden

- static SessionManager getInstance ()
- static void invalidateUserSession (String userId)

1.10.32.2 Dokumentation der Elementfunktionen

static SessionManager getInstance () [static]

Liefert den Session Manager zurück.

Rückgabe:

Definiert in Zeile 46 der Datei SessionManager.java.

Benutzt SessionManager.instance.

static void invalidateUserSession (String userId) [static]

bricht eine Sitzung ab.

Parameter:

userId

Definiert in Zeile 55 der Datei SessionManager.java.

void valueBound (HttpSessionBindingEvent event)

Diese Methode wird aufgerufen, wenn sich ein Nutzer einloggt.

Definiert in Zeile 69 der Datei SessionManager.java.

void valueUnbound (HttpSessionBindingEvent event)

Diese Methode wird aufgerufen, wenn sich ein Nutzer ausloggt.

Definiert in Zeile 94 der Datei SessionManager.java.

Benutzt SysMessage.setPayload(), SysMessage.setReceiver(), SysMessage.toString() und SysEvent.toString().

1.10.33 SysEvent Klassenreferenz

Basisklasse für ModuleEvent, Notification und SessionClosed.

1.10.33.1 Ausführliche Beschreibung

Diese Klasse wird benutzt, wenn ein Ereignis im EAM-System ausgelöst wird, wie z.b. Ein Modul installiert oder entfernt sowie eine Systemnachricht gesendet.

Autor:

Yu

Definiert in Zeile 16 der Datei SysEvent.java.

Öffentliche Methoden

- final String get (String key)
- final String getDescription ()
- final String getName ()

- final String getType ()
- final void put (String key, String value)
- final void setDescription (String description)
- final void setName (String name)
- final void setType (String type)
- final String toString ()

Statische öffentliche Attribute

- static final String MODULE_SYS_EVENT_TYPE = "MODULE_SYS_EVENT_TYPE"
- static final String NOTIFICATION_SYS_EVENT_TYPE = "NOTIFICATION_SYS_EVENT_TYPE"
- static final String SESSION_CLOSED_SYS_EVENT_TYPE = "SESSION_CLOSED_SYS_EVENT_TYPE"
- static final String SYS_EVENT_DESCRIPTION = "SYS_EVENT_DESCRIPTION"
- static final String SYS_EVENT_NAME = "SYS_EVENT_NAME"
- static final String SYS_EVENT_TITEL = "SYS_EVENT_TITEL"
- static final String SYS_EVENT_TYPE = "SYS_EVENT_TYPE"

1.10.33.2 Dokumentation der Elementfunktionen

final String get (String key)

Liefert den Wert eines Parameters zurück.

Parameter:

key der Name des Parameters

Rückgabe:

der Wert des Parameters

Definiert in Zeile 121 der Datei SysEvent.java.

final String getDescription ()

Liefert die Beschreibung eines Ereignis zurück.

Rückgabe:

die Beschreibung eines Ereignis

Definiert in Zeile 79 der Datei SysEvent.java.

final String getName ()

Liefert den Namen eines Ereignis zurück.

Rückgabe:

der Namen eines Ereignis

Definiert in Zeile 95 der Datei SysEvent.java.

final String getType ()

Liefert den Typ eines Ereignis zurück.

Rückgabe:

der Typ eines Ereignis

Definiert in Zeile 63 der Datei SysEvent.java.

final void put (String key, String value)

Weist einem Parameter einen Wert zu.

Parameter:

key der Name des Parameters

value der Wert

Definiert in Zeile 112 der Datei SysEvent.java.

Wird benutzt von ModuleEvent.ModuleEvent(), ModuleEvent.setAction(), ModuleEvent.setBundleId() und ModuleEvent.setBundleName().

final void setDescription (String description)

Setter der Beschreibung eines Ereignis

Parameter:

description die neue Beschreibung eines Ereignis

Definiert in Zeile 87 der Datei SysEvent.java.

final void setName (String name)

Setter des Namens eines Ereignis

Parameter:

name der neue Name eines Ereignis

Definiert in Zeile 103 der Datei SysEvent.java.

final void setType (String type)

Setter des Typs eines Ereignis

Parameter:

type der neue Typ

Definiert in Zeile 71 der Datei SysEvent.java.

Wird benutzt von ModuleEvent.ModuleEvent().

final String toString ()

Konvertiert ein SysEvent zu einem String

Definiert in Zeile 129 der Datei SysEvent.java.

Benutzt PrintfFormat.sprintf().

Wird benutzt von SysMsgManager.notification() und SessionManager.valueUnbound().

1.10.33.3 Dokumentation der Datenelemente

final String MODULE_SYS_EVENT_TYPE = "MODULE_SYS_EVENT_TYPE"
[static]

Konstante für den Typ eines Modulereignis

Definiert in Zeile 26 der Datei SysEvent.java.

final String NOTIFICATION_SYS_EVENT_TYPE =
"NOTIFICATION_SYS_EVENT_TYPE" [static]

Konstante für den Typ einer Systemmitteilung

Definiert in Zeile 31 der Datei SysEvent.java.

final String SESSION_CLOSED_SYS_EVENT_TYPE =
"SESSION_CLOSED_SYS_EVENT_TYPE" [static]

Konstante für den Typ eines Sessionablaufereignis

Definiert in Zeile 36 der Datei SysEvent.java.

final String SYS_EVENT_DESCRIPTION = "SYS_EVENT_DESCRIPTION"
[static]

Konstante für die Systemereignisbeschreibung

Definiert in Zeile 46 der Datei SysEvent.java.

final String SYS_EVENT_NAME = "SYS_EVENT_NAME" [static]

Konstante für den Systemereignisnamen

Definiert in Zeile 41 der Datei SysEvent.java.

final String SYS_EVENT_TITEL = "SYS_EVENT_TITEL" [static]

Konstante für den Systemereignistitel

Definiert in Zeile 51 der Datei SysEvent.java.

final String SYS_EVENT_TYPE = "SYS_EVENT_TYPE" [static]

Konstante für ein Systemereignis

Definiert in Zeile 21 der Datei SysEvent.java.

1.10.34 SysMessage Klassenreferenz

1.10.34.1 Ausführliche Beschreibung

Diese Klasse wird benutzt ein Systemereignis zu einem String zu konvertieren.

Autor:

Yu

Definiert in Zeile 12 der Datei SysMessage.java.

Öffentliche Methoden

- String getPayload ()
- Map< String, String > getReceiver ()
- void setPayload (String payload)
- void setReceiver (Map< String, String > receiver)
- String toString ()

Statische öffentliche Attribute

- static final String PARAM_FORMATTER = "%s=%s"
- static final String PARAM_SEPERATOR = ";"

1.10.34.2 Dokumentation der Elementfunktionen

String getPayload ()

Liefert den Inhalt der Nachricht zurück.

Rückgabe:

Definiert in Zeile 56 der Datei SysMessage.java.

Map<String, String> getReceiver ()

Liefert die Empfänger zurück.

Rückgabe:

ein Map für die Empfänger

Definiert in Zeile 40 der Datei SysMessage.java.

void setPayload (String payload)

Setter des Inhalts der Nachricht.

Parameter:

payload

Definiert in Zeile 64 der Datei SysMessage.java.

Wird benutzt von SysMsgManager.notification() und SessionManager.valueUnbound().

void setReceiver (Map< String, String > receiver)

Setter der Empfänger

Parameter:

receiver die neue Empfänger

Definiert in Zeile 48 der Datei SysMessage.java.

Wird benutzt von SysMsgManager.notification() und SessionManager.valueUnbound().

String toString ()

Konvertiert eine Nachricht zu einem String

Definiert in Zeile 71 der Datei SysMessage.java.

Benutzt PrintfFormat.sprintf().

Wird benutzt von SessionManager.valueUnbound().

1.10.34.3 Dokumentation der Datenelemente

final String PARAM_FORMATTER = "%s=%s" [static]

formatter

Definiert in Zeile 17 der Datei SysMessage.java.

final String PARAM_SEPERATOR = ";" [static]

das Trennungszeichen für Parameter

Definiert in Zeile 22 der Datei SysMessage.java.

1.10.35 SysMsgServerThread Klassenreferenz

1.10.35.1 Ausführliche Beschreibung

Diese Klasse wird benutzt, wenn der SysMsgManager eine Systemnachricht an einem Nutzer senden möchte.

Autor:

Yu

Definiert in Zeile 16 der Datei SysMsgServerThread.java.

Öffentliche Methoden

- void run ()

1.10.35.2 Dokumentation der Elementfunktionen

void run ()

die Nachricht wird hier gesendet.

Definiert in Zeile 41 der Datei SysMsgServerThread.java.

1.10.36 TopMenu Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::api::AbstractAdoptableMenuContributor und de::offis::pg::eam::core::menu::api::ITopMenuConstants.

1.10.36.1 Ausführliche Beschreibung

Diese Klasse wird dazu benutzt, um der Kern seine Menü (Topmenu) anzumelden.

Autor:

Yu

Definiert in Zeile 21 der Datei TopMenu.java.

Öffentliche Methoden

- synchronized void addMenuContributor (IMenuContributor menuContributor)
- boolean addMenuUpdateListener (ITopMenuUpdateListener listener)
- String getAdopterId ()

- String getContributionTarget ()
- String getContributorId ()
- String getTargetSubmenu ()
- synchronized String getTopMenuParams (Set< Pair > views)
- synchronized void removeMenuContributor (IMenuContributor menuContributor)
- boolean removeMenuUpdateListener (ITopMenuUpdateListener listener)
- TopMenu ()

Öffentliche, statische Methoden

- static TopMenu getInstance ()

Geschützte Methoden

- void activate (ComponentContext context)
- void deactivate (ComponentContext context) throws Exception

1.10.36.2 Beschreibung der Konstruktoren und Destruktoren

TopMenu ()

Konstruktor

Definiert in Zeile 46 der Datei TopMenu.java.

Benutzt TopMenu.addMenuUpdateListener() und
AbstractMenuContributor.initMenuContributorFromDefaultConfigXML().

1.10.36.3 Dokumentation der Elementfunktionen

void activate (ComponentContext context) [protected]

Diese Methode wird von OSGI aufgerufen, wenn der Contributor erfolgreich startet wird.

Parameter:

context der Bundlekontext

Definiert in Zeile 151 der Datei TopMenu.java.

synchronized void addMenuContributor (IMenuContributor menuContributor)

Fügt einen neuen Contributor hinzu.

Erneute Implementation von AbstractAdoptableMenuContributor.

Definiert in Zeile 55 der Datei TopMenu.java.

boolean addMenuUpdateListener (ITopMenuUpdateListener listener)

Fügt einen neuen Listener hinzu.

Parameter:

listener ein neuer Listener

Rückgabe:

Definiert in Zeile 70 der Datei TopMenu.java.

Wird benutzt von TopMenu.TopMenu().

void deactivate (ComponentContext context) throws Exception [protected]

Diese Methode wird von OSGI aufgerufen, wenn der Contributor gestoppt wird.

Parameter:

context der Bundlekontext

Ausnahmebehandlung:**Exception**

Definiert in Zeile 160 der Datei TopMenu.java.

String getAdopterId ()

Liefert die ID des Adopter zurück (Topmenu ist sowohl ein Adopter als auch ein Contributor).

Implementiert IMenuContributorAdopter.

Definiert in Zeile 78 der Datei TopMenu.java.

String getContributionTarget ()

Liefert Beitragsziel zurück.

Implementiert IMenuContributor.

Definiert in Zeile 85 der Datei TopMenu.java.

String getContributorId ()

Liefert die ID des Contributors zurück (Topmenu ist sowohl ein Adopter als auch ein Contributor).

Implementiert IMenuContributor.

Definiert in Zeile 93 der Datei TopMenu.java.

static TopMenu getInstance () [static]

Liefer das Topmenu zurück.

Rückgabe:

Definiert in Zeile 34 der Datei TopMenu.java.

Benutzt TopMenu.instance.

String getTargetSubmenu ()

Topmenu trägt seine Menü nur zum TopToolBar. Deswegen liefert diese Methode null zurück.

Implementiert IMenuContributor.

Definiert in Zeile 101 der Datei TopMenu.java.

synchronized String getTopMenuParams (Set< Pair > views)

Liefert die Parameter des Topmenu zurück.

Parameter:

views die Views eines Nutzers

Rückgabe:

Menüparameter als String

Definiert in Zeile 110 der Datei TopMenu.java.

Benutzt Submenu.accept(), AbstractMenuContributor.getMenuItems() und JddmMenuParamsGenerator.getMenuParams().

synchronized void removeMenuContributor (IMenuContributor menuContributor)

Entfernt ein Contributor.

Erneute Implementation von AbstractAdoptableMenuContributor.

Definiert in Zeile 127 der Datei TopMenu.java.

boolean removeMenuUpdateListener (ITopMenuUpdateListener listener)

Entfernt einen Listener

Parameter:

listener der zu entfernende Listener

Rückgabe:

Definiert in Zeile 143 der Datei TopMenu.java.

1.10.37 TopMenuBar Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::menu::api::AbstractMenuContributorAdopter`.

1.10.37.1 Ausführliche Beschreibung

Diese Klasse wird dazu benutzt, um die Menüeinträge von Topmenu aufzunehmen.

Autor:

Yu

Definiert in Zeile 17 der Datei TopMenuBar.java.

Statische öffentliche Attribute

- `static final String TOP_MENU_BAR = "TOP_MENU_BAR"`

Geschützte Methoden

- `void activate (ComponentContext context)`
- `void deactivate (ComponentContext context) throws Exception`

1.10.37.2 Dokumentation der Elementfunktionen

void activate (ComponentContext context) [protected]

Diese Methode wird von OSGI aufgerufen, wenn der Adopter erfolgreich startet wird.

Parameter:

context der Bundlekontext

Definiert in Zeile 38 der Datei TopMenuBar.java.

void deactivate (ComponentContext context) throws Exception [protected]

Diese Methode wird von OSGI aufgerufen, wenn der Adopter gestoppt wird.

Parameter:

context der Bundlekontext

Ausnahmebehandlung:

Exception

Definiert in Zeile 50 der Datei TopMenuBar.java.

1.10.37.3 Dokumentation der Datenelemente

final String TOP_MENU_BAR = "TOP_MENU_BAR" [static]

Konstante für TopMenuBar

Definiert in Zeile 22 der Datei TopMenuBar.java.

1.10.38 TopMenuServerThread Klassenreferenz

1.10.38.1 Ausführliche Beschreibung

Dies Thread wird dazu benutzt, wenn das Topmenu aktualisiert wird und die erneute Menüparameter an Clients gesendet werden sollen.

Autor:

Yu

Definiert in Zeile 29 der Datei TopMenuServerThread.java.

Öffentliche Methoden

- void run ()
- TopMenuServerThread (Socket socket, String userId, int appletId)

1.10.38.2 Beschreibung der Konstruktoren und Destruktoren

TopMenuServerThread (Socket socket, String userId, int appletId)

Konstruktor

Parameter:

socket die Socketverbindung

userId die Nutzerid

appletId die Appletid

Definiert in Zeile 63 der Datei TopMenuServerThread.java.

Benutzt UMUser.getViews().

1.10.38.3 Dokumentation der Elementfunktionen

void run ()

In dieser Methode wird die erneute Menüparameter an Clients gesendet.

Definiert in Zeile 136 der Datei TopMenuServerThread.java.

1.11 metamodel

1.11.1 CheckRights Klassenreferenz

1.11.1.1 Ausführliche Beschreibung

Stellt Methoden zur Prüfung von Rechten bereit.

Autor:

Roland Koppe

Definiert in Zeile 23 der Datei CheckRights.java.

Öffentliche, statische Methoden

- static boolean checkRights (String instanceName, HashMap< String, Long > rights, Right[] neededRights)
- static boolean checkRights (EAMAttribute attr, HashMap< String, Long > rights, Right[] neededRights)
- static boolean checkRights (EAMSuperClass objectRelation, HashMap< String, Long > rights, Right[] neededRights)

1.11.1.2 Dokumentation der Elementfunktionen

static boolean checkRights (String instanceName, HashMap< String, Long > rights, Right[] neededRights) [static]

Prüft, ob die Rechte für eine Instanz-Tabelle vorhanden sind.

Parameter:

instanceName String Name der Instanz-Tabelle

rights HashMap<String, Long>

neededRights Array von AuthNames.Right

Rückgabe:

true, wenn Rechte vorhanden

Definiert in Zeile 168 der Datei CheckRights.java.

Benutzt CheckRights.checkRights().

static boolean checkRights (EAMAttribute attr, HashMap< String, Long > rights, Right[] neededRights) [static]

Prüft, ob die gewünschten Rechte für das angegebene EAMAttribute vorhanden sind.

Parameter:

attr EAMAttribute
rights HashMap<String, Long>
neededRights Array von AuthNames.Right

Rückgabe:

true, wenn Rechte vorhanden

Definiert in Zeile 108 der Datei CheckRights.java.

static boolean checkRights (EAMSuperClass objectRelation, HashMap< String, Long > rights, Right[] neededRights) [static]

Prüft, ob die gewünschten Rechte für das angebene EAMObjekt / EAMRelation existieren.

Parameter:

objectRelation EAMObject / EAMRelation
rights HashMap<String, Long>
neededRights Array von AuthNames.Right

Rückgabe:

true, wenn Rechte vorhanden

Definiert in Zeile 35 der Datei CheckRights.java.

Wird benutzt von CheckRights.checkRights().

1.11.2 Validator Klassenreferenz

1.11.2.1 Ausführliche Beschreibung

Valididator für Attribute und DatenTypen.

Autor:

Roland Koppe
David Heimann

Definiert in Zeile 37 der Datei metamodel/datatype/model/Validator.java.

Öffentliche, statische Methoden

- static DataType dataTypeFromAttribute (EAMAttribute attr)
- static boolean isBinary (EAMAttribute attr)
- static boolean isBoolean (EAMAttribute attr)
- static boolean isChar (EAMAttribute attr)

- static boolean isDate (EAMAttribute attr)
- static boolean isList (EAMAttribute attr)
- static boolean isNumeric (EAMAttribute attr)
- static boolean isNumericUnsigned (EAMAttribute attr)
- static boolean isValidAttribute (EAMAttribute attr) throws InstanceException
- static boolean isValidValue (EAMAttribute attr, Object value) throws InstanceException

1.11.2.2 Dokumentation der Elementfunktionen

static DataType dataTypeFromAttribute (EAMAttribute attr) [static]

Liefert den DataType zum angegebenen Attribut.

Parameter:

attr EAMAttribute

Rückgabe:

Definiert in Zeile 47 der Datei metamodel/datatype/model/Validator.java.

Wird benutzt von Validator.isBinary(), Validator.isBoolean(), Validator.isChar(), Validator.isDate(), Validator.isList(), Validator.isNumeric(), Validator.isNumericUnsigned(), Validator.isValidAttribute() und Validator.isValidValue().

static boolean isBinary (EAMAttribute attr) [static]

Liefert zurück, ob das angegebene Attribut von einem binären Datentyp ist.

Parameter:

attr EAMAttribute

Rückgabe:

true, wenn Attribut von einem binären Datentyp

Definiert in Zeile 125 der Datei metamodel/datatype/model/Validator.java.

Benutzt Validator.dataTypeFromAttribute().

static boolean isBoolean (EAMAttribute attr) [static]

Liefert zurück, ob das angegebene Attribut von einem boolschen Datentyp ist.

Parameter:

attr EAMAttribute

Rückgabe:

true, wenn Attribut von einem boolschen Datentyp

Definiert in Zeile 101 der Datei metamodel/datatype/model/Validator.java.

Benutzt Validator.dataTypeFromAttribute().

static boolean isChar (EAMAttribute attr) [static]

Liefert zurück, ob das angegebene Attribut von einem Text-Datentyp ist.

Parameter:

attr EAMAttribute

Rückgabe:

true, wenn Attribut von einem Text-Datentyp

Definiert in Zeile 137 der Datei metamodel/datatype/model/Validator.java.

Benutzt Validator.dataTypeFromAttribute().

static boolean isDate (EAMAttribute attr) [static]

Liefert zurück, ob das angegebene Attribut von einem Datum-Datentyp ist.

Parameter:

attr EAMAttribute

Rückgabe:

true, wenn Attribut von einem Datum-Datentyp

Definiert in Zeile 113 der Datei metamodel/datatype/model/Validator.java.

Benutzt Validator.dataTypeFromAttribute().

static boolean isList (EAMAttribute attr) [static]

Liefert zurück, ob das angegebene Attribut von einem Listen-Datentyp ist.

Parameter:

attr EAMAttribute

Rückgabe:

true, wenn Attribut von einem Listen-Datentyp

Definiert in Zeile 149 der Datei metamodel/datatype/model/Validator.java.

Benutzt Validator.dataTypeFromAttribute().

static boolean isNumeric (EAMAttribute attr) [static]

Liefert zurück, ob das angegebene Attribut von einem numerischen Datentyp ist.

Parameter:

attr EAMAttribute

Rückgabe:

true, wenn Attribut vom numerischen Datentyp

Definiert in Zeile 76 der Datei metamodel/datatype/model/Validator.java.

Benutzt Validator.dataTypeFromAttribute().

static boolean isNumericUnsigned (EAMAttribute attr) [static]

Liefert zurück, ob das angegebene Attribut von einem nicht vorzeichenbehafteten numerischen Datentyp ist.

Parameter:

attr EAMAttribute

Rückgabe:

true, wenn Attribut vom nicht vorzeichenbehafteten numerischen Datentyp

Definiert in Zeile 89 der Datei metamodel/datatype/model/Validator.java.

Benutzt Validator.dataTypeFromAttribute().

static boolean isValidAttribute (EAMAttribute attr) throws InstanceException [static]

Prüft, ob das angegebene Attribut korrekt zum Datentyp des Attributs passt

Parameter:

attr EAMAttribute

Rückgabe:

true, wenn Attribut zum Datentyp passt

Ausnahmebehandlung:

InstanceException

Definiert in Zeile 418 der Datei metamodel/datatype/model/Validator.java.

Benutzt Validator.dataTypeFromAttribute().

static boolean isValidValue (EAMAttribute attr, Object value) throws InstanceException [static]

Prüft, ob der angegebene Wert um angegebenen Attribut passt. Dabei wird zunächst versucht den Wert in den Attribut-Typ zu casten. Dann folgen die Prüfungen der Einschränkungen, wie min/max, defaultValue, regulären Ausdruck.

Parameter:

attr EAMAttribute

value Object Wert

Rückgabe:

true, wenn Wert zum Attribut passt

Ausnahmebehandlung:

InstanceAttributeException

Definiert in Zeile 166 der Datei metamodel/datatype/model/Validator.java.

Benutzt Validator.dataTypeFromAttribute().

1.11.3 InstanceAttributeDataTypeDomainException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceAttributeException.

1.11.3.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 13 der Datei InstanceAttributeDataTypeDomainException.java.

1.11.4 InstanceAttributeDataTypeMissingException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceAttributeException.

1.11.4.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 13 der Datei InstanceAttributeDataTypeMissingException.java.

1.11.5 InstanceAttributeDomainException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceAttributeException.

1.11.5.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceAttributeDomainException.java.

1.11.6 InstanceAttributeEnumDomainException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceAttributeException.

1.11.6.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceAttributeEnumDomainException.java.

1.11.7 InstanceAttributeException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceException.

Basisklasse für InstanceAttributeDataTypeDomainException, InstanceAttributeDataTypeMissingException, InstanceAttributeDomainException, InstanceAttributeEnumDomainException, InstanceAttributeNotUniqueValueException, InstanceAttributeRegexException, InstanceAttributeTypeDomainException, InstanceAttributeTypeMissingException, InstanceAttributeValueIncompatibleException und InstanceAttributeValueNotNumericException.

1.11.7.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceAttributeException.java.

1.11.8 InstanceAttributeNotUniqueValueException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceAttributeException.

1.11.8.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 13 der Datei InstanceAttributeNotUniqueValueException.java.

1.11.9 InstanceAttributeRegexException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceAttributeException.

1.11.9.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceAttributeRegexException.java.

1.11.10 InstanceAttributeTypeDomainException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceAttributeException.

1.11.10.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceAttributeTypeDomainException.java.

1.11.11 InstanceAttributeTypeMissingException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceAttributeException.

1.11.11.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceAttributeTypeMissingException.java.

1.11.12 InstanceAttributeValueIncompatibleException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceAttributeException.

1.11.12.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceAttributeValueIncompatibleException.java.

1.11.13 InstanceAttributeValueNotNumericException Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::metamodel::exceptions::InstanceAttributeException`.

1.11.13.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei `InstanceAttributeValueNotNumericException.java`.

1.11.14 InstanceDatabaseException Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::metamodel::exceptions::InstanceException`.

1.11.14.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei `InstanceDatabaseException.java`.

1.11.15 InstanceDataObjectDeleteException Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::metamodel::exceptions::InstanceException`.

1.11.15.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei `InstanceDataObjectDeleteException.java`.

1.11.16 InstanceDataObjectSaveException Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::metamodel::exceptions::InstanceException`.

1.11.16.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei `InstanceDataObjectSaveException.java`.

1.11.17 InstanceException Klassenreferenz

Basisklasse für InstanceAttributeException, InstanceDatabaseException, InstanceDataObjectDeleteException, InstanceDataObjectSaveException, InstanceIsNullException, InstanceLockingException, InstanceMetaObjectException, InstanceRegexException, InstanceRelationException, InstanceRightsMissingException, InstanceSQLInvalidException, InstanceSQLTableNotExistsException, InstanceTableDeleteException, InstanceTableException und InstanceTableSaveException.

1.11.17.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceException.java.

1.11.18 InstanceIsNullException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceException.

1.11.18.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceIsNullException.java.

1.11.19 InstanceLockingException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceException.

Basisklasse für InstanceLockingInUseException.

1.11.19.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceLockingException.java.

1.11.20 InstanceLockingInUseException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceLockingException.

1.11.20.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceLockingInUseException.java.

1.11.21 InstanceMetaObjectException Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::metamodel::exceptions::InstanceException`.

Basisklasse für `InstanceMetaObjectHasNoAttributesException`,
`InstanceMetaObjectInvalidException` und `InstanceMetaObjectIsNullException`.

1.11.21.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceMetaObjectException.java.

1.11.22 InstanceMetaObjectHasNoAttributesException Klassenreferenz

Abgeleitet von

`de::offis::pg::eam::core::metamodel::exceptions::InstanceMetaObjectException`.

1.11.22.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceMetaObjectHasNoAttributesException.java.

1.11.23 InstanceMetaObjectInvalidException Klassenreferenz

Abgeleitet von

`de::offis::pg::eam::core::metamodel::exceptions::InstanceMetaObjectException`.

1.11.23.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceMetaObjectInvalidException.java.

1.11.24 InstanceMetaObjectIsNullException Klassenreferenz

Abgeleitet von

de::offis::pg::eam::core::metamodel::exceptions::InstanceMetaObjectException.

1.11.24.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceMetaObjectIsNullException.java.

1.11.25 InstanceRegexException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceException.

1.11.25.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceRegexException.java.

1.11.26 InstanceRelationException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceException.

Basisklasse für InstanceRelationFirstInvalidException, InstanceRelationFirstMissingException, InstanceRelationFirstMultiplicityException, InstanceRelationMultiplicityException, InstanceRelationSecondInvalidException, InstanceRelationSecondMissingException und InstanceRelationSecondMultiplicityException.

1.11.26.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceRelationException.java.

1.11.27 InstanceRelationFirstInvalidException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceRelationException.

1.11.27.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 11 der Datei InstanceRelationFirstInvalidException.java.

1.11.28 InstanceRelationFirstMissingException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceRelationException.

1.11.28.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceRelationFirstMissingException.java.

1.11.29 InstanceRelationFirstMultiplicityException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceRelationException.

1.11.29.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceRelationFirstMultiplicityException.java.

1.11.30 InstanceRelationMultiplicityException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceRelationException.

1.11.30.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceRelationMultiplicityException.java.

1.11.31 InstanceRelationSecondInvalidException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceRelationException.

1.11.31.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 11 der Datei InstanceRelationSecondInvalidException.java.

1.11.32 InstanceRelationSecondMissingException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceRelationException.

1.11.32.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceRelationSecondMissingException.java.

1.11.33 InstanceRelationSecondMultiplicityException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceRelationException.

1.11.33.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceRelationSecondMultiplicityException.java.

1.11.34 InstanceRightsMissingException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceException.

1.11.34.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceRightsMissingException.java.

1.11.35 InstanceSQLInvalidException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceException.

1.11.35.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceSQLException.java.

1.11.36 InstanceSQLTableNotExistsException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceException.

1.11.36.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceSQLTableNotExistsException.java.

1.11.37 InstanceTableAttributeDataTypeDomainException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceTableException.

1.11.37.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 13 der Datei InstanceTableAttributeDataTypeDomainException.java.

1.11.38 InstanceTableAttributeDataTypeMissingException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceTableException.

1.11.38.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceTableAttributeDataTypeMissingException.java.

1.11.39 InstanceTableAttributeDefaultValueIncompatibleException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceTableException.

1.11.39.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei

InstanceTableAttributeDefaultValueIncompatibleException.java.

1.11.40 InstanceTableAttributeIncompatibleException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceTableException.

1.11.40.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceTableAttributeIncompatibleException.java.

1.11.41 InstanceTableAttributeTypeMissingException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceTableException.

1.11.41.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceTableAttributeTypeMissingException.java.

1.11.42 InstanceTableDeleteException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceException.

1.11.42.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceTableDeleteException.java.

1.11.43 InstanceTableEnumRegexException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceTableException.

1.11.43.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceTableEnumRegexException.java.

1.11.44 InstanceTableEnumRegexMissingException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceTableException.

1.11.44.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei InstanceTableEnumRegexMissingException.java.

1.11.45 InstanceTableException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceException.

Basisklasse für InstanceTableAttributeDataTypeDomainException,
InstanceTableAttributeDataTypeMissingException,
InstanceTableAttributeDefaultValueIncompatibleException,
InstanceTableAttributeIncompatibleException,
InstanceTableAttributeTypeMissingException, InstanceTableEnumRegexException und
InstanceTableEnumRegexMissingException.

1.11.45.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceTableException.java.

1.11.46 InstanceTableSaveException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::exceptions::InstanceException.

1.11.46.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei InstanceTableSaveException.java.

1.11.47 DataObjectList Klassenreferenz

1.11.47.1 Ausführliche Beschreibung

Diese Klasse stellt eine dynamische Liste von DataObjects bereit. Bei Bedarf werden weitere DataObjects aus der Datenbank nachgeladen. Nach der Verwendung ist die DataObjectList mit close zu beenden.

Autor:

Roland Koppe

Definiert in Zeile 21 der Datei DataObjectList.java.

Öffentliche Methoden

- void close ()
- DataObjectList (IInstanceLoader loader) throws SQLException, DatabaseException
- int getAbsoulteIndex ()
- int getCurrentPage ()
- ArrayList< DataObject > getDataObjects ()
- int getIndex ()
- IInstanceLoader getInstanceLoader ()
- int getListSize ()
- int getPages ()
- void move (int page) throws SQLException, DatabaseException
- void moveFirst () throws SQLException, DatabaseException
- void moveLast () throws SQLException, DatabaseException
- void moveNext () throws SQLException, DatabaseException
- void movePrevious () throws SQLException, DatabaseException
- DataObject next () throws SQLException, DatabaseException
- DataObject previous () throws SQLException, DatabaseException
- void reset (IInstanceLoader loader) throws SQLException, DatabaseException
- int size ()

1.11.47.2 Beschreibung der Konstruktoren und Destruktoren

DataObjectList (IInstanceLoader loader) throws SQLException, DatabaseException

Konstruktor erzeugt eine neue DataObjectList der angegebenen Größe. Zum Laden der DataObjects muss eine InstanceLoader Instanz angegeben werden.

Parameter:

loader InstanceLoader

size int Seiten-Größe

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 41 der Datei DataObjectList.java.

Benutzt DataObjectList.reset().

1.11.47.3 Dokumentation der Elementfunktionen**void close ()**

Beendet den Zugriff auf die Datenbank.

Definiert in Zeile 317 der Datei DataObjectList.java.

int getAbsoulteIndex ()

Liefert den Index des DataObjects über alle DataObjects.

Rückgabe:

int Index des DataObjects

Definiert in Zeile 236 der Datei DataObjectList.java.

int getCurrentPage ()

Liefert die aktuelle Seite.

Rückgabe:

int aktuelle Seite

Definiert in Zeile 285 der Datei DataObjectList.java.

Wird benutzt von DataObjectScrollerTag.doStartTag().

ArrayList<DataObject> getDataObjects ()

Liefert die aktuellen DataObjects der Seite als ArrayList.

Rückgabe:

ArrayList<DataObject>

Definiert in Zeile 297 der Datei DataObjectList.java.

int getIndex ()

Liefert den aktuellen Index des DataObjects auf der Seite.

Rückgabe:

int Index des DataObjects

Definiert in Zeile 224 der Datei DataObjectList.java.

InstanceLoader getInstanceLoader ()

Liefert den InstanceLoader.

Rückgabe:

InstanceLoader

Definiert in Zeile 308 der Datei DataObjectList.java.

int getListSize ()

Liefert die Größe der Seite, d.h. die Anzahl der DataObjects je Seite.

Rückgabe:

int Seitengröße

Definiert in Zeile 252 der Datei DataObjectList.java.

int getPages ()

Liefert die Anzahl der Seiten.

Rückgabe:

int Seitenzahl

Definiert in Zeile 263 der Datei DataObjectList.java.

Wird benutzt von DataObjectScrollerTag.doStartTag().

void move (int page) throws SQLException, DatabaseException

Kehrt auf die angegebene Seite und lädt die DataObjects.

Parameter:

page int Seite

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 193 der Datei DataObjectList.java.

Wird benutzt von DataObjectTableTag.doScrollerAction().

void moveFirst () throws SQLException, DatabaseException

Kehrt auf die erste Seite und lädt die DataObjects.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 140 der Datei DataObjectList.java.

Wird benutzt von DataObjectTableTag.doScrollerAction().

void moveLast () throws SQLException, DatabaseException

Kehrt auf die letzte Seite und lädt die DataObjects.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 209 der Datei DataObjectList.java.

Wird benutzt von DataObjectTableTag.doScrollerAction().

void moveNext () throws SQLException, DatabaseException

Kehrt auf die nächste Seite und lädt die DataObjects.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 176 der Datei DataObjectList.java.

Wird benutzt von DataObjectTableTag.doScrollerAction().

void movePrevious () throws SQLException, DatabaseException

Kehrt auf die vorherige Seite und lädt die DataObjects.

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 160 der Datei DataObjectList.java.

Wird benutzt von DataObjectTableTag.doScrollerAction().

DataObject next () throws SQLException, DatabaseException

Holt das nächste DataObject aus der DataObjectList und setzt den Zeiger auf das nächste DataObject.

Rückgabe:

DataObject

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 79 der Datei DataObjectList.java.

DataObject previous () throws SQLException, DatabaseException

Holt das vorherige DataObject aus der DataObjectList und setzt den Zeiger auf das vorhergenede DataObject.

Rückgabe:

DataObject

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 109 der Datei DataObjectList.java.

void reset (InstanceLoader loader) throws SQLException, DatabaseException

Initialisiert die DataObjectList und setzt den Zeiger auf das erste Element.

Parameter:

loader InstanceLoader

size int Seiten-Größe

Ausnahmebehandlung:

SQLException

DatabaseException

Definiert in Zeile 56 der Datei DataObjectList.java.

Wird benutzt von DataObjectList.DataObjectList().

int size ()

Liefert die Anzahl der DataObjects für die gesamte Anfrage.

Rückgabe:

int Anzahl der DataObjects

Definiert in Zeile 274 der Datei DataObjectList.java.

1.11.48 InstanceDataInterface Schnittstellenreferenz**1.11.48.1 Ausführliche Beschreibung**

AUTO-CREATED INTERFACE FOR
de.offis.pg.eam.core.metamodel.instance.api.InstanceDataInterface

Definiert in Zeile 13 der Datei IInstanceDataInterface.java.

1.11.49 InstanceQuery Schnittstellenreferenz**1.11.49.1 Ausführliche Beschreibung**

AUTO-CREATED INTERFACE FOR
de.offis.pg.eam.core.metamodel.instance.api.InstanceQuery

Definiert in Zeile 17 der Datei IInstanceQuery.java.

1.11.50 InstanceDataInterface Klassenreferenz**1.11.50.1 Ausführliche Beschreibung**

Erlaubt die Manipulation von Instanzen eines Metamodells.

Autor:

Roland Koppe

Definiert in Zeile 50 der Datei InstanceDataInterface.java.

Öffentliche, statische Methoden

- static boolean deleteObject (DataObject dataObject, String userId, String password)
throws InstanceException, InstanceRightsMissingException
- static long saveObject (DataObject dataObject, String userId, String password)
throws InstanceException, InstanceRightsMissingException

1.11.50.2 Dokumentation der Elementfunktionen

static boolean deleteObject (DataObject dataObject, String userId, String password) throws InstanceException, InstanceRightsMissingException [static]

Löscht die Instanz (DataObject) eines EAMObject oder einer EAMRelation.

Parameter:

dataObject DataObject
userId String userId
password String Passwort des Users

Rückgabe:

true, wenn erfolgreich

Ausnahmebehandlung:

InstanceException Fehler beim Löschen der Instanz - MataObject prüfen
InstanceMissingRights Keine Rechte für das Löschen vorhanden

Definiert in Zeile 417 der Datei InstanceDataInterface.java.

static long saveObject (DataObject dataObject, String userId, String password) throws InstanceException, InstanceRightsMissingException [static]

Speichert die angegebene Instanz (DataObject) eines EAMObjects oder einer EAMRelation und gibt deren id zurück. Soll ein DataObject aktualisiert in der Datenbank gespeichert werden, so muss die ID des DataObject gesetzt sein.

Parameter:

dataObject DataObject
userId String userId
password String password des Users

Rückgabe:

long id der Instanz

Ausnahmebehandlung:

InstanceException Hinweise auf Fehler beim speichern der Instanz
DatabaseException
InstanceMissingRights

Definiert in Zeile 67 der Datei InstanceDataInterface.java.

1.11.51 InstanceInterface Klassenreferenz

1.11.51.1 Ausführliche Beschreibung

Hilfsmethoden für InstanceDataInterface bzw. InstanceMetaInterface.

Autor:

Roland Koppe

Definiert in Zeile 17 der Datei InstanceInterface.java.

Öffentliche, statische Methoden

- static String getAttributeIdByColumnName (String name)
- static String getColumnNameByAttribute (EAMAttribute attr)
- static String getColumnShortNameByAttribute (EAMAttribute attr)
- static String getInstanceTableName (EAMSuperClass objectRelation)
- static String relaxAttributeName (String attrName)

1.11.51.2 Dokumentation der Elementfunktionen

static String getAttributeIdByColumnName (String name) [static]

Liefert die zum Attribut gehörende ID anhand des Spaltennamens in einer Instanz-Tabelle.

Parameter:

name String

Rückgabe:

String der ID, null wenn nicht erfolgreich

Definiert in Zeile 92 der Datei InstanceInterface.java.

static String getColumnNameByAttribute (EAMAttribute attr) [static]

Liefert den zum Attribut gehörenden Namen von Spalten in Instanz-Tabellen von Objekten und Relationen.

Parameter:

attr EAMAttribute

Rückgabe:

String Name des Attributs

Definiert in Zeile 69 der Datei InstanceInterface.java.

Benutzt InstanceInterface.relaxAttributeName().

static String getColumnShortNameByAttribute (EAMAttribute attr) [static]

Liefert den zum Attribut gehörenden Namen von Spalten in Instanz-Tabellen von Objekten und Relationen.

Parameter:

attr EAMAttribute

Rückgabe:

String Name des Attributs

Definiert in Zeile 32 der Datei InstanceInterface.java.

static String getInstanceTableName (EAMSuperClass objectRelation) [static]

Liefert den Namen der Instanztabelle zum angegebenen Objekt.

Parameter:

EAMSuperClass EAMObject oder EAMRelation

Definiert in Zeile 111 der Datei InstanceInterface.java.

static String relaxAttributeName (String attrName) [static]

Liefert einen "relaxten" Attributnamen zum angegebenen Namen zurück. Ein relaxter Name enthält keine Sonderzeichen und ist max 20 Zeichen lang.

Parameter:

attrName Name des Attributs

Rückgabe:

String relaxter Name des Attributes

Definiert in Zeile 49 der Datei InstanceInterface.java.

Wird benutzt von InstanceInterface.getColumnNameByAttribute().

1.11.52 InstanceLoader Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::instance::impl::IInstanceLoader.

1.11.52.1 Ausführliche Beschreibung

Der InstanceLoader ermöglicht das Laden von DataObjects aus der Datenbank.

Autor:

Roland Koppe

Definiert in Zeile 34 der Datei InstanceLoader.java.

Öffentliche Methoden

- `InstanceLoader (EAMSuperClass objectRelation, long id, String userId, String password, int fetchSize)` throws `InstanceException`

1.11.52.2 Beschreibung der Konstruktoren und Destruktoren

InstanceLoader (EAMSuperClass objectRelation, long id, String userId, String password, int fetchSize) throws InstanceException

Konstruktor. Bereitet die Anfrage von `DataObjects` vor. Prüft die Rechte des Benutzers.

Parameter:

objectRelation EAMSuperClass EAMObject / EAMRelation
id long ID der Instanz, wenn 0, dann alle Instanzen
userId String ID des Benutzers
password String Passwort des Benutzers

Ausnahmebehandlung:

InstanceException

Definiert in Zeile 65 der Datei `InstanceLoader.java`.

1.11.53 InstanceLoaderSQL Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::metamodel::instance::impl::IInstanceLoader`.

1.11.53.1 Ausführliche Beschreibung

Der `InstanceLoader` ermöglicht das Laden von `DataObjects` aus der Datenbank.

Autor:

Roland Koppe

Definiert in Zeile 30 der Datei `InstanceLoaderSQL.java`.

Öffentliche Methoden

- `InstanceLoaderSQL (String sql, String userId, String password, int fetchSize)` throws `InstanceException`

1.11.53.2 Beschreibung der Konstruktoren und Destruktoren

InstanceLoaderSQL (String sql, String userId, String password, int fetchSize) throws InstanceException

Konstruktor. Bereitet die Anfrage von DataObjects vor. Prüft die Rechte des Benutzers.

Parameter:

objectRelation EAMSuperClass EAMObject / EAMRelation

id long ID der Instanz, wenn 0, dann alle Instanzen

userId String ID des Benutzers

password String Passwort des Benutzers

Ausnahmebehandlung:

InstanceException

Definiert in Zeile 63 der Datei InstanceLoaderSQL.java.

Benutzt Query.getSQLView() und Query.parse().

1.11.54 InstanceMetaInterface Klassenreferenz

1.11.54.1 Ausführliche Beschreibung

Erlaubt die Manipulation von Instanz-Tabellen von EAMObjects und EAMRelations.

Autor:

Roland Koppe

Definiert in Zeile 36 der Datei InstanceMetaInterface.java.

Öffentliche, statische Methoden

- static synchronized boolean deleteObject (EAMSuperClass objectRelation) throws DatabaseException
- static synchronized long saveObjectX (EAMSuperClass objectRelation) throws InstanceException

1.11.54.2 Dokumentation der Elementfunktionen

static synchronized boolean deleteObject (EAMSuperClass objectRelation) throws DatabaseException [static]

Löscht die Instanz-Tabelle des angegebenen EAMObject / EAMRelation.

Parameter:

objectRelation EAMObject / EAMRelation

Rückgabe:

true, wenn erfolgreich

Ausnahmebehandlung:**DatabaseException**

Definiert in Zeile 346 der Datei InstanceMetaInterface.java.

static synchronized long saveObjectX (EAMSuperClass objectRelation) throws InstanceException [static]

Legt die Instanz-Tabelle für ein Objekt bzw. eine Relation an oder editiert diese Instanz-Tabelle wie im EAMObject bzw. EAMRelation angegeben ist.

Parameter:

objectRelation EAMObject / EAMRelation

Rückgabe:

id des Objekts/Relation

Ausnahmebehandlung:**InstanceException**

Definiert in Zeile 49 der Datei InstanceMetaInterface.java.

Benutzt Column.setAllowNull(), Column.setDefaultValue(), Column.setField() und Column.setType().

1.11.55 InstanceQuery Klassenreferenz**1.11.55.1 Ausführliche Beschreibung****Autor:**

Roland Koppe

Definiert in Zeile 35 der Datei InstanceQuery.java.

Öffentliche, statische Methoden

- static DataObject queryDataObject (EAMSuperClass objectRelation, long id, String userId, String password) throws InstanceException, DatabaseException
- static ArrayList< DataObject > queryDataObjects (String sql, String userId, String password) throws InstanceException, DatabaseException
- static ArrayList< DataObject > queryDataObjects (EAMSuperClass objectRelation, long id, String userId, String password) throws InstanceException
- static ArrayList< DataObject > queryDataObjects (EAMSuperClass objectRelation, String userId, String password) throws InstanceException

1.11.55.2 Dokumentation der Elementfunktionen

static DataObject queryDataObject (EAMSuperClass objectRelation, long id, String userId, String password) throws InstanceException, DatabaseException [static]

Liefert die Instanz mit der angegebenen ID zum angegebenen MetaObject.

Parameter:

objectRelation EAMObject / EAMRelation

id long ID der Instanz

userId String ID des Users

password String Passwort des Users

Rückgabe:

DataObject, wenn eines gefunden wurde, sonst null

Ausnahmebehandlung:

InstanceException

DatabaseException

Definiert in Zeile 466 der Datei InstanceQuery.java.

static ArrayList<DataObject> queryDataObjects (String sql, String userId, String password) throws InstanceException, DatabaseException [static]

Liefert alle Instanz-Objekte zu der angegebenen SQL Anfrage. Sollen nur Instanzen eines EAMObject oder einer EAMRelation abgefragt werden, so sollte folgende Methode verwendet werden.

Siehe auch:

```
public static ArrayList<DataObject> queryDataObjects(EAMSuperClass  
objectRelation, String userId, String password) throws InstanceException
```

Werden mehrere Instanzen miteinander verknüpft, etwa per JOIN, so sind die Attribute eines DataObject wie folgt definiert:

- DataObject.getId() liefert -1
- DataObject.getMetaObject() liefert null
- label alle Label des Objektes
- alle normalen Attribute nur über ihre ID
- zusammengesetzte oder umbenannte Attribute werden mit dem Prefix _ gesetzt
- IDs über
 - idobj_<id> liefert die ID der Instanz des EAMObject mit der ID <id>
 - idrel_<id> liefert die ID der Instanz der EAMRelation mit der ID <id>

- `idobj_first_<id>` liefert die ID der ersten, an der Relation mit der ID `<id>` beteiligten, Instanz eines `EAMObject`.
- `idobj_second_<id>` liefert die ID der zweiten, an der Relation mit der ID `<id>` beteiligten, Instanz eines `EAMObject`.

Parameter:

sql
userId
password

Rückgabe:**Ausnahmebehandlung:**

InstanceException
DatabaseException

Definiert in Zeile 507 der Datei `InstanceQuery.java`.

static ArrayList<DataObject> queryDataObjects (EAMSuperClass objectRelation, long id, String userId, String password) throws InstanceException [static]

Liefert alle Instanz-Objekte zum angegebenen `EAMObject` / `EAMRelation`. Das `DataObject` enthält ein Attribut "id" welches die ID der Instanz für das jeweilige `EAMObject` / `EAMRelation` beschreibt. Handelt es sich bei der Angabe um eine `EAMRelation`, so werden zusätzlich noch die Attribute "idobj_first" und "idobj_second" bereitgestellt, die auf die IDs der Objekt-Instanzen verweisen. Wird der Parameter `id` mit `> 0` angegeben, so wird die Instanz mit der ID `id` zurückgegeben.

Parameter:

objectRelation `EAMObject` oder `EAMRelation`
id long ID der Instanz, `<= 0`, dann alle Instanzen
userId String ID des Users
password String Passwort des Users

Rückgabe:

`ArrayList<DataObject>` Liste von `DataObjects`

Ausnahmebehandlung:

InstanceException
DatabaseException

Definiert in Zeile 79 der Datei `InstanceQuery.java`.

static ArrayList<DataObject> queryDataObjects (EAMSuperClass objectRelation, String userId, String password) throws InstanceException [static]

Liefert alle Instanz-Objekte zum angegebenen EAMObject / EAMRelation. Das DataObject enthält ein Attribut "id" welches die ID der Instanz für das jeweilige EAMObject / EAMRelation beschreibt. Handelt es sich bei der Angabe um eine EAMRelation, so werden zusätzlich noch die Attribute "idobj_first" und "idobj_second" bereitgestellt, die auf die IDs der Objekt-Instanzen verweisen.

Parameter:

objectRelation EAMObject oder EAMRelation

userId String ID des Users

password String Passwort des Users

Rückgabe:

ArrayList<DataObject> Liste von DataObjects

Ausnahmebehandlung:

InstanceException

DatabaseException

Definiert in Zeile 52 der Datei InstanceQuery.java.

1.11.56 Query Klassenreferenz

1.11.56.1 Ausführliche Beschreibung

Query ermöglicht das Parsen einer SQL Anfrage (SELECT) und entsorgt Felder (Attribute), die mit den angegebenen Rechten eines Benutzers nicht eingesehen werden können.

Die Standard-Attribute wie "id", "idobj_first" und "idobj_second" können dabei weiterhin von allen Benutzern gesehen werden.

bekannte Probleme:

- Tabellen Aliase werden in SELECT nicht beachtet Lösung: vollständigen Tabellennamen verwenden

Autor:

Roland Koppe

Definiert in Zeile 33 der Datei instance/impl/Query.java.

Öffentliche Methoden

- ArrayList< String > getFields ()

- String getSQL ()
- String getSQLView ()
- ArrayList< String > getTables ()
- void parse (String sql, HashMap< String, Long > rights) throws InstanceSQLTableNotExistsException, DatabaseException
- void parse (HashMap< String, Long > rights) throws InstanceSQLTableNotExistsException, DatabaseException
- Query (String sql)
- Query ()
- void reset ()

1.11.56.2 Beschreibung der Konstruktoren und Destruktoren

Query ()

Standard-Konstruktor ohne Wirkung.

Definiert in Zeile 183 der Datei instance/impl/Query.java.

Query (String sql)

Konstruktor. Erwartet die zu parsende SQL Anfrage.

Parameter:

sql String

Definiert in Zeile 194 der Datei instance/impl/Query.java.

1.11.56.3 Dokumentation der Elementfunktionen

ArrayList<String> getFields ()

Liefert alle aus der SQL Anfrage geparsten Feldnamen. Es werden auch solche Feldnamen geliefert, für die keine Rechte bestehen.

Rückgabe:

ArrayList<String> Liste von Feldnamen

Definiert in Zeile 707 der Datei instance/impl/Query.java.

String getSQL ()

Liefert die originale SQL Anfrage.

Rückgabe:

String SQL Anfrage

Definiert in Zeile 731 der Datei instance/impl/Query.java.

String getSQLView ()

Liefert die geparste Version der SQL Anfrage.

Rückgabe:

String SQL Anfrage

Definiert in Zeile 741 der Datei instance/impl/Query.java.

Wird benutzt von InstanceLoaderSQL.InstanceLoaderSQL().

ArrayList<String> getTables ()

Liefert alle aus der SQL Anfrage geparsten Tabellennamen. Es werden auch solche Feldnamen geliefert, für die keine Rechte bestehen.

Rückgabe:

ArrayList<String> Liste von Tabellennamen

Definiert in Zeile 720 der Datei instance/impl/Query.java.

void parse (String sql, HashMap< String, Long > rights) throws InstanceSQLTableNotExistsException, DatabaseException

Parst die angegeben SQL Anfrage mit den angegebenen Rechten.

Siehe auch:

```
public void parse(HashMap<String, Long> rights) throws  
InstanceSQLTableNotExistsException
```

Parameter:

sql String SQL Anfrage

rights HashMap<String, Long> Liste von Rechten

Ausnahmebehandlung:

InstanceSQLTableNotExistsException

DatabaseException

Definiert in Zeile 597 der Datei instance/impl/Query.java.

Benutzt Query.parse() und Query.reset().

void parse (HashMap< String, Long > rights) throws InstanceSQLTableNotExistsException, DatabaseException

Parst die zuvor angegebene SQL Anfrage mit den hier angegebenen Rechten. Dabei werden Felder im SELECT Teil nach Feldern mit Rechten gefiltert, so dass nur Felder gezeigt werden, für die auch Rechte existieren. Wurde das Parsen erfolgreich beendet kann die erzeugte neue SQL Anweisung mit `getSQLView()` geholt werden.

Parameter:

rights `HashMap<String, Long>` Liste von Rechten

Ausnahmebehandlung:

InstanceSQLTableNotExistsException

DatabaseException

Definiert in Zeile 317 der Datei `instance/impl/Query.java`.

Wird benutzt von `InstanceLoaderSQL.InstanceLoaderSQL()` und `Query.parse()`.

void reset ()

Setzt das Query-Objekt zurück.

Definiert in Zeile 750 der Datei `instance/impl/Query.java`.

Wird benutzt von `Query.parse()`.

1.11.57 Lock Klassenreferenz

1.11.57.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 14 der Datei `Lock.java`.

Öffentliche Methoden

- `long getInstanceId ()`
- `Date getLockDate ()`
- `EAMSuperClass getMetaObject ()`
- `long getUserId ()`
- `Lock (EAMSuperClass metaObject, long instanceId, long userId)`

1.11.57.2 Beschreibung der Konstruktoren und Destruktoren

Lock (EAMSuperClass metaObject, long instanceId, long userId)

Konstruktor für ein neues Lock-Objekt, welches Informationen über einen Lock speichert.

Parameter:

metaObject EAMObject / EAMRelation

instanceld long Id der Instanz

userId long Id des Users

Definiert in Zeile 33 der Datei Lock.java.

1.11.57.3 Dokumentation der Elementfunktionen

long getInstanceld ()

Liefert die Id der Instanz dieses Locks.

Rückgabe:

long Id der Instanz

Definiert in Zeile 60 der Datei Lock.java.

Date getLockDate ()

Liefert das Datum des Locks.

Rückgabe:

Date

Definiert in Zeile 82 der Datei Lock.java.

EAMSuperClass getMetaObject ()

Liefert das MetaObject dieses Locks.

Rückgabe:

EAMSuperClass

Definiert in Zeile 49 der Datei Lock.java.

long getUserId ()

Liefert die Id des Users.

Rückgabe:

long Id des Users

Definiert in Zeile 71 der Datei Lock.java.

Wird benutzt von Locking.getUserLocks() und Locking.killUser().

1.11.58 Locking Klassenreferenz

1.11.58.1 Ausführliche Beschreibung

Diese Klasse ermöglicht das Locking von Instanzen von Metamodellen. Das Locking bleibt nur für die Laufzeit des Systems erhalten.

Autor:

Roland Koppe

Definiert in Zeile 21 der Datei Locking.java.

Öffentliche, statische Methoden

- static ArrayList< Lock > getUserLocks (long userId)
- static synchronized Lock isLocked (EAMSuperClass metaObject, long instanceId)
- static synchronized void killUser (long userId)
- static synchronized Lock lock (EAMSuperClass metaObject, long instanceId, long userId) throws InstanceLockingException
- static synchronized void reset ()
- static synchronized boolean unlock (Lock lock)

1.11.58.2 Dokumentation der Elementfunktionen

static ArrayList<Lock> getUserLocks (long userId) [static]

Liefert die von dem angegebenen Benutzer gesperrten Objekte als Liste von Locks.

Parameter:

userId long Id des Users

Rückgabe:

ArrayList<Lock> Liste von Sperrungen

Definiert in Zeile 142 der Datei Locking.java.

Benutzt Lock.getUserId().

static synchronized Lock isLocked (EAMSuperClass metaObject, long instanceId) [static]

Liefert zurück, ob die angegebene Instanz des MetaObjekts bereits gesperrt ist. Wenn bereits eine Sperrung vorliegt, werden die Lock-Daten zurückgegeben.

Parameter:

metaObject EAMObject / EAMRelation

instanceld long Id der Instanz

Rückgabe:

Lock, wenn vorhanden, sonst null

Definiert in Zeile 71 der Datei Locking.java.

static synchronized void killUser (long userId) [static]

Löscht alle zum angegebenen Benutzer gehörenden Sperrungen. Dies kann hilfreich sein, wenn ein Benutzer "gleichzeitig" mehrere Objekte bearbeiten möchte oder der Benutzer vom System abgemeldet wird.

Parameter:

userId long Id des Users

Definiert in Zeile 168 der Datei Locking.java.

Benutzt Lock.getUserId().

static synchronized Lock lock (EAMSuperClass metaObject, long instanceld, long userId) throws InstanceLockingException [static]

Liefert ein Lock auf die angegebene Instanz des MetaObject mit der angegebenen Instanz-Id. Es wird eine InstanceLockingException geworfen, wenn bereits ein Lock auf die angegebene Instanz vorliegt.

Parameter:

metaObject EAMObject / EAMRelation

instanceld long Id der Instanz

userId long id des Users

Rückgabe:

Lock, wenn erfolgreich, sonst null bzw. Exception

Ausnahmebehandlung:

InstanceLockingException

Definiert in Zeile 94 der Datei Locking.java.

static synchronized void reset () [static]

Löst alle Locks auf.

Definiert in Zeile 192 der Datei Locking.java.

static synchronized boolean unlock (Lock lock) [static]

Hebt die angegebene Sperrung für eine Instanz wieder auf.

Parameter:

lock Lock

Rückgabe:

true, wenn erfolgreich und Sperrung existierte

Definiert in Zeile 122 der Datei Locking.java.

1.11.59 DataObject Klassenreferenz

1.11.59.1 Ausführliche Beschreibung

Ein DataObject stellt eine Instanz eines EAMObject oder einer EAMRelation dar und besitzt Informationen über das entsprechende EAMObject bzw. EAMRelation. Außerdem enthält es eine HashMap mit Schlüssel-Wert-Paaren, die diese Instanz ausmachen.

Autor:

Roland Koppe

Definiert in Zeile 25 der Datei DataObject.java.

Öffentliche Methoden

- void addAttribute (String key, Object value)
- void clearAttributes ()
- void delAttribute (String key)
- HashMap< String, Object > getAttributes ()
- long getId ()
- String getLabel ()
- HashMap< String, Object > getLabels ()
- EAMSuperClass getMetaObject ()
- boolean isObject ()
- boolean isRelation ()
- void setAttributes (HashMap< String, Object > attributes)
- void setId (long id)
- void setLabels (HashMap< String, Object > labels)
- void setMetaObject (Object metaObject)
- String toString ()

1.11.59.2 Dokumentation der Elementfunktionen

void addAttribute (String key, Object value)

Fügt den Attributen des DataObject ein weiteres hinzu.

Siehe auch:

```
public HashMap<String, Object> getAttributes()
```

Parameter:

key String Schlüssel ID

value Object beliebiger Wert

Definiert in Zeile 134 der Datei DataObject.java.

void clearAttributes ()

Leert die Liste von Attributen des DataObject. Es werden KEINE Werte aus der Datenbank gelöscht oder die Werte auf NULL gesetzt.

Definiert in Zeile 154 der Datei DataObject.java.

void delAttribute (String key)

Löscht das angegebene Attribut des DataObjects, indem es auf null gesetzt wird.

Parameter:

key String Schlüssel ID

Definiert in Zeile 144 der Datei DataObject.java.

HashMap<String, Object> getAttributes ()

Liefert alle Attribute des DataObject als HashMap zurück. Der Key ist dabei ein String und der Wert ein beliebiges Objekt. Key ist immer die ID des Attributs. Es existieren vier Spezialfälle: id, die ID des DataObject idobj_first, für EAMRelation die erste beteiligte EAMObject Instanz idobj_second, für EAMRelation die zweite beteiligte EAMObject Instanz label, der Anzeigename

Rückgabe:

```
HashMap<String, Object>
```

Definiert in Zeile 113 der Datei DataObject.java.

long getId ()

Liefert die ID des DataObject.

Rückgabe:

long ID

Definiert in Zeile 39 der Datei DataObject.java.

String getLabel ()

Liefert das Label des DataObjects. Besitzt das DataObject mehrere Label, so werden sie durch Kommata getrennt.

Rückgabe:

String label

Definiert in Zeile 166 der Datei DataObject.java.

HashMap<String, Object> getLabels ()

Liefert alle Labels des DataObject als HashMap mit der Attribut-ID als Schlüssel und das eigentliche Label als Wert.

Rückgabe:

HashMap<String, Object>

Definiert in Zeile 197 der Datei DataObject.java.

EAMSuperClass getMetaObject ()

Liefert das MetaObject des DataObject. Es sollte ein EAMObject oder eine EAMRelation sein.

Rückgabe:

EAMSuperClass

Definiert in Zeile 60 der Datei DataObject.java.

boolean isObject ()

Liefert true, wenn es sich bei dem MetaObject des DataObject um ein EAMObject handelt.

Rückgabe:

true, wenn MetaObject EAMObject

Definiert in Zeile 86 der Datei DataObject.java.

boolean isRelation ()

Liefert true, wenn es sich bei dem MetaObject des DataObject um eine EAMRelation handelt.

Rückgabe:

true, wenn MetaObject EAMRelation

Definiert in Zeile 96 der Datei DataObject.java.

void setAttributes (HashMap< String, Object > attributes)

Setzt alle Attribute des DataObject.

Siehe auch:

public HashMap<String, Object> getAttributes()

Parameter:

attributes HashMap<String, Object>

Definiert in Zeile 123 der Datei DataObject.java.

void setId (long id)

Setzt die ID des DataObject.

Parameter:

id long ID

Definiert in Zeile 48 der Datei DataObject.java.

void setLabels (HashMap< String, Object > labels)

Setzt das Label des DataObjects.

Achtung: Diese Methode dient ist nur für die Anfrage von DataObjects gedacht. Änderungen der Labels haben keine Auswirkungen auf den Datenbestand. Zum Ändern eines Attributs (welches auch ein Label sein kann) ist geeignet public HashMap<String, Object> getAttributes() oder public void addAttribute(String key, Object value) zu verwenden.

Parameter:

labels HashMap<String, Object> mit Label-ID, Value

Definiert in Zeile 215 der Datei DataObject.java.

void setMetaObject (Object metaObject)

Setzt das MetaObject des DataObject. Das MetaObject kann ein EAMObject oder eine EAMRelation sein.

Parameter:

metaObject Object (EAMObject oder EAMRelation)

Definiert in Zeile 70 der Datei DataObject.java.

String toString ()

Liefert die Attribute des DataObject als String zurück. Dabei werden die einzelnen Attribute durch Kommata getrennt.

Definiert in Zeile 225 der Datei DataObject.java.

1.11.60 DataObjectBean Klassenreferenz**1.11.60.1 Ausführliche Beschreibung**

Einfache Test-Bean um das DataObjectTableTag zu demonstrieren.

Autor:

Roland Koppe

Definiert in Zeile 18 der Datei DataObjectBean.java.

1.11.61 DataObjectScrollerTag Klassenreferenz

Abgeleitet von javax::servlet::jsp::tagext::TagSupport.

1.11.61.1 Ausführliche Beschreibung

Das DataObjectScrollerTag stellt einen Scroller für dargestellte DataObjects durch das DataObjectTableTag dar.

Autor:

Roland Koppe

Definiert in Zeile 21 der Datei DataObjectScrollerTag.java.

Öffentliche Methoden

- int doStartTag () throws JspException
- DataObjectList getDataObjectList ()
- String getRedirect ()
- String getScrollerId ()
- void setDataObjectList (DataObjectList dataObjectList)

- void setIconFirst (String iconFirst)
- void setIconLast (String iconLast)
- void setIconNext (String iconNext)
- void setIconPrevious (String iconPrevious)
- void setPageContext (PageContext pageContext)
- void setRedirect (String redirectURI)
- void setScrollerId (String id)

1.11.61.2 Dokumentation der Elementfunktionen

int doStartTag () throws JspException

Schreibt den Scroller.

Definiert in Zeile 173 der Datei DataObjectScrollerTag.java.

Benutzt DataObjectList.getCurrentPage() und DataObjectList.getPages().

DataObjectList getDataObjectList ()

Liefert die DataObjectList des Scrollers.

Rückgabe:

DataObjectList

Definiert in Zeile 72 der Datei DataObjectScrollerTag.java.

String getRedirect ()

Liefert die URL des Redirects, nach dem Betätigen eines Knopfs.

Rückgabe:

String URI

Definiert in Zeile 138 der Datei DataObjectScrollerTag.java.

String getScrollerId ()

Liefert die HTML-ID des Scrollers.

Rückgabe:

String ID des Scrollers

Definiert in Zeile 50 der Datei DataObjectScrollerTag.java.

void setDataObjectList (DataObjectList dataObjectList)

Setzt die `DataObjectList` für den Scroller.

Parameter:

`dataObjectList` `DataObjectList`

Definiert in Zeile 83 der Datei `DataObjectScrollerTag.java`.

`void setIconFirst (String iconFirst)`

Setzt die URL für das Bild von "ID_FIRST".

Parameter:

`iconFirst` `String` URL

Definiert in Zeile 94 der Datei `DataObjectScrollerTag.java`.

`void setIconLast (String iconLast)`

Setzt die URL für das Bild von "ID_LAST".

Parameter:

`iconPrevious` `String` URL

Definiert in Zeile 127 der Datei `DataObjectScrollerTag.java`.

`void setIconNext (String iconNext)`

Setzt die URL für das Bild von "ID_NEXT".

Parameter:

`iconNext` `String` URL

Definiert in Zeile 116 der Datei `DataObjectScrollerTag.java`.

`void setIconPrevious (String iconPrevious)`

Setzt die URL für das Bild von "ID_PREVIOUS".

Parameter:

`iconPrevious` `String` URL

Definiert in Zeile 105 der Datei `DataObjectScrollerTag.java`.

`void setPageContext (PageContext pageContext)`

Gibt den JavaScript-Ausdruck für das onclick-Ereignis der Knöpfe zurück.

Parameter:

`name` `String` ID Namen, Konstanten

Rückgabe:

String onclick JavaScript Gibt den JavaScript-Ausdruck für das onclick-Ereignis der Seitenzahl zurück.

Parameter:

page int Seitenzahl

Rückgabe:

String onclick JavaScript

Definiert in Zeile 321 der Datei DataObjectScrollerTag.java.

void setRedirect (String redirectURI)

Setzt die URL für den Redirect.

Parameter:

redirectURI String URI

Definiert in Zeile 149 der Datei DataObjectScrollerTag.java.

void setScrollerId (String id)

Setzt die HTML-ID des Scrollers.

Parameter:

id String ID des Scrollers

Definiert in Zeile 61 der Datei DataObjectScrollerTag.java.

1.11.62 DataObjectTableTag Klassenreferenz

Abgeleitet von javax::servlet::jsp::tagext::TagSupport.

1.11.62.1 Ausführliche Beschreibung

Dieses Tag zeigt eine DataObjectList in einer Tabelle an.

Autor:

Roland Koppe

Definiert in Zeile 29 der Datei DataObjectTableTag.java.

Öffentliche Methoden

- DataObjectList getDataObjectList ()

- String getScrollerId ()
- String getTableId ()
- void setDataObjectList (DataObjectList dataObjectList)
- void setScrollerId (String id)
- void setTableId (String id)

Geschützte Methoden

- void doScrollerAction (String scrollerAction, String scrollerPage)

1.11.62.2 Dokumentation der Elementfunktionen

void doScrollerAction (String scrollerAction, String scrollerPage) [protected]

Führt eine Scroller-Aktion auf der DataObjectList aus.

Siehe auch:

DataObjectScrollerTag

Definiert in Zeile 126 der Datei DataObjectTableTag.java.

Benutzt DataObjectList.move(), DataObjectList.moveFirst(), DataObjectList.moveLast(), DataObjectList.moveNext() und DataObjectList.movePrevious().

DataObjectList getDataObjectList ()

Liefert die DataObjectList.

Rückgabe:

DataObjectList

Definiert in Zeile 46 der Datei DataObjectTableTag.java.

String getScrollerId ()

Liefert den HTML-ID des Scrollers. Optional.

Rückgabe:

String ID des Scrollers

Definiert in Zeile 90 der Datei DataObjectTableTag.java.

String getTableId ()

Liefert die HTML-ID der Tabelle.

Rückgabe:

String ID der Tabelle

Definiert in Zeile 68 der Datei DataObjectTableTag.java.

void setDataObjectList (DataObjectList dataObjectList)

Setzt die DataObjectList.

Parameter:

dataObjectList DataObjectList

Definiert in Zeile 57 der Datei DataObjectTableTag.java.

void setScrollerId (String id)

Setzt die HTML-ID des Scrollers. Optional.

Parameter:

id String ID des Scrollers

Definiert in Zeile 101 der Datei DataObjectTableTag.java.

void setTableId (String id)

Setzt die HTML-ID der Tabelle.

Parameter:

id String ID der Tabelle

Definiert in Zeile 79 der Datei DataObjectTableTag.java.

1.11.63 DAOFactory Klassenreferenz

Basisklasse für HibernateDAOFactoryImpl.

1.11.63.1 Ausführliche Beschreibung

Factory um DAOs zu erzeugen. Anhand der DAOs kann ein Zugriff auf die gewünschten Entitäten erfolgen.

Die DAOs dürfen nur innerhalb einer Transaktion verwendet werden.

Dies ist durch die Verwendung des Servletfilters HibernateTransactionServletfilter möglich - dort sind auch Methoden, welche eine manuelle Transaktionsdemarkation erlauben.

Definiert in Zeile 28 der Datei /factory/DAOFactory.java.

Öffentliche Methoden

- abstract EAMAttributeDAO getEAMAttributeDAO ()
- abstract EAMAttributeTypeDAO getEAMAttributeTypeDAO ()
- abstract EAMCategoryDAO getEAMCategoryDAO ()
- abstract EAMObjectDAO getEAMObjectDAO ()
- abstract EAMObjectTypeDAO getEAMObjectTypeDAO ()
- abstract EAMRelationDAO getEAMRelationDAO ()
- abstract EAMRelationTypeDAO getEAMRelationTypeDAO ()
- abstract EAMStatusDAO getEAMStatusDAO ()
- abstract EAMTimeDAO getEAMTimeDAO ()
- abstract EAMTypeDAO getEAMTypeDAO ()

Öffentliche, statische Methoden

- static DAOFactory getAuthInstance (String userid, String password)

Statische öffentliche Attribute

- static final Class HIBERNATE = HibernateDAOFactoryImpl.class

Geschützte, statische Methoden

- static DAOFactory instance (Class factory)

1.11.63.2 Dokumentation der Elementfunktionen

static DAOFactory getAuthInstance (String userid, String password) [static]

Siehe auch:

DAOFactory.getAuthInstance(String, String) Instanziert eine DAOFactory mit den entsprechenden Benutzerdaten

Parameter:

userid Id des Benutzers

password Passwort des Benutzers

Rückgabe:

DAOFactory mit den entsprechenden Benutzerinformationen zur Überprüfung der Rechte

Definiert in Zeile 73 der Datei /factory/DAOFactory.java.

Benutzt DAOFactory.HIBERNATE, DAOFactory.instance() und DAOFactory.setCredentials().

abstract EAMAttributeDAO getEAMAttributeDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMAttribute

abstract EAMAttributeTypeDAO getEAMAttributeTypeDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMAttributeType

abstract EAMCategoryDAO getEAMCategoryDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMCategory

abstract EAMObjectDAO getEAMObjectDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMObject

abstract EAMObjectTypeDAO getEAMObjectTypeDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMObjectType

abstract EAMRelationDAO getEAMRelationDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMRelation

abstract EAMRelationTypeDAO getEAMRelationTypeDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMRelationType

abstract EAMStatusDAO getEAMStatusDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMStatus

abstract EAMTimeDAO getEAMTimeDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMTime

abstract EAMTypeDAO getEAMTypeDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMType

static DAOFactory instance (Class factory) [static, protected]

Fabrikmethode zur Instanziierung einer speziellen Fabrik

Definiert in Zeile 48 der Datei /factory/DAOFactory.java.

Wird benutzt von DAOFactory.getAuthInstance().

1.11.63.3 Dokumentation der Datenelemente

final Class HIBERNATE = HibernateDAOFactoryImpl.class [static]

Erstellt eine alleinstehende DAOFactory, die "unmanaged" DAO beans zurückgibt die in jeder Umgebung für die Hibernate konfiguriert wurde benutzt werden kann. Benutzt HibernateUtil/AbstractSessionFactory und Hibernate context propagation (CurrentSessionContext), Thread-gebunden oder Transaktion- gebunden und Transaktions-gültig

Definiert in Zeile 43 der Datei /factory/DAOFactory.java.

Wird benutzt von DAOFactory.getAuthInstance().

1.11.64 HibernateDAOFactoryImpl Klassenreferenz

Abgeleitet von

de::offis::pg::eam::core::metamodel::metainterface::dao2::factory::DAOFactory.

1.11.64.1 Ausführliche Beschreibung

Implementierung der DAOFactory mit Hibernate

Diese Klasse enthält ferner einige private Klassen, die DAO Implementierungen darstellen und direkt von GenericDAO abgeleitet sind

Siehe auch:

 DAOFactory

Definiert in Zeile 41 der Datei /factory/HibernateDAOFactoryImpl.java.

1.11.65 EAMObjectDAOHibernate Klassenreferenz

Abgeleitet von

de::offis::pg::eam::core::metamodel::metainterface::dao::impl::GenericHibernateDAOImpl< de::offis::pg::eam::core::metamodel::metadata::EAMObject, Long > und
de::offis::pg::eam::core::metamodel::metainterface::dao2::interfaces::EAMObjectDAO.

1.11.65.1 Ausführliche Beschreibung

Implementierung einer DAO für EAMObject

Definiert in Zeile 16 der Datei /impl/EAMObjectDAOHibernate.java.

Öffentliche Methoden

- List< EAMObject > findByCategory (EAMCategory cat)

1.11.65.2 Dokumentation der Elementfunktionen

List<EAMObject> findByCategory (EAMCategory cat)

Sucht alle EAMObject Objekte in einer angegebenen Kategorie

Parameter:

cat Zu durchsuchende Kategorie

Rückgabe:

 Liste aller EAMObject Objekte, die in der EAMCategory cat sind

Implementiert EAMObjectDAO.

Definiert in Zeile 26 der Datei /impl/EAMObjectDAOHibernate.java.

1.11.66 EAMRelationDAOHibernate Klassenreferenz

Abgeleitet von

de::offis::pg::eam::core::metamodel::metainterface::dao::impl::GenericHibernateDAOImpl<

de::offis::pg::eam::core::metamodel::metadata::EAMRelation, Long > und
de::offis::pg::eam::core::metamodel::metainterface::dao2::interfaces::EAMRelationDAO.

1.11.66.1 Ausführliche Beschreibung

Implementierung einer DAO für Relationen

Definiert in Zeile 16 der Datei /impl/EAMRelationDAOHibernate.java.

Öffentliche Methoden

- List< EAMRelation > findByCategory (EAMCategory cat)
- List< EAMRelation > findByObject (EAMObject eamobject)

1.11.66.2 Dokumentation der Elementfunktionen

List<EAMRelation> findByCategory (EAMCategory cat)

Sucht alle EAMRelation Objekte in einer angegebenen Kategorie

Parameter:

cat Zu durchsuchende Kategorie

Rückgabe:

Liste aller EAMRelation Objekte, die in der EAMCategory cat sind

Implementiert EAMRelationDAO.

Definiert in Zeile 28 der Datei /impl/EAMRelationDAOHibernate.java.

List<EAMRelation> findByObject (EAMObject eamobject)

Sucht alle an einer EAMRelation Objekte, an denen eamobject beteiligt ist

Siehe auch:

EAMObject.getRelations()

Rückgabe:

Liste der EAMRelation an denen eamobject beteiligt ist

Implementiert EAMRelationDAO.

Definiert in Zeile 41 der Datei /impl/EAMRelationDAOHibernate.java.

1.11.67 GenericHibernateDAOImpl< T, ID extends Serializable > Klassenreferenz

Abgeleitet von

de::offis::pg::eam::core::metamodel::metainterface::dao::interfaces::GenericDAO< T, ID >.

1.11.67.1 Ausführliche Beschreibung

Implementierung eines generischen DAO mit Hiberate

Parameter:

<T> Typ der zu speichernden Klasse

<ID> Typ der Id (des Primärschlüssels)

Definiert in Zeile 36 der Datei /impl/GenericHibernateDAOImpl.java.

Öffentliche Methoden

- void delete (T entity)
- List< T > findAll ()
- List< T > findByCriteria (Criterion...criterion)
- List< T > findByExample (T exampleInstance, String...excludeProperty)
- T findById (ID id, boolean lock)
- Class< T > getPersistentClass ()
- Session getSession ()
- T save (T entity) throws Exception
- void setCredentials (String userid, String password)
- void setSession (Session s)

Geschützte Methoden

- void filterListByReadPermission (List< T > list)

1.11.67.2 Dokumentation der Elementfunktionen

void delete (T entity)

Siehe auch:

GenericDAO.delete()

Definiert in Zeile 245 der Datei /impl/GenericHibernateDAOImpl.java.

void filterListByReadPermission (List< T > list) [protected]

Entfernt aus einer übergebenen Liste die Elemente, für die kein Leserecht besteht

Parameter:

list Liste die "gefiltert" werden soll

Definiert in Zeile 355 der Datei /impl/GenericHibernateDAOImpl.java.

List<T> findAll ()

Siehe auch:

GenericDAO.findAll()

Definiert in Zeile 154 der Datei /impl/GenericHibernateDAOImpl.java.

List<T> findByCriteria (Criterion... criterion)

Methode für Hibernate Criteria Queries

Parameter:

criterion Hibernate Criterions

Rückgabe:

Liste mit Ergebnissen der Anfrage

Definiert in Zeile 337 der Datei /impl/GenericHibernateDAOImpl.java.

List<T> findByExample (T exampleInstance, String... excludeProperty)

Siehe auch:

GenericDAO.findByExample()

Definiert in Zeile 163 der Datei /impl/GenericHibernateDAOImpl.java.

T findById (ID id, boolean lock)

Siehe auch:

GenericDAO.findById()

Definiert in Zeile 129 der Datei /impl/GenericHibernateDAOImpl.java.

Class<T> getPersistentClass ()

Rückgabe:

Klasse/Type die durch die DAO persistiert wird

Definiert in Zeile 120 der Datei /impl/GenericHibernateDAOImpl.java.

Session getSession ()

Rückgabe:

Die Hibernate Session der DAO

Definiert in Zeile 102 der Datei /impl/GenericHibernateDAOImpl.java.

T save (T entity) throws Exception

Siehe auch:

GenericDAO.save()

Definiert in Zeile 184 der Datei /impl/GenericHibernateDAOImpl.java.

void setCredentials (String userid, String password)

Methode zum Setzen der Zugangsdaten; Initialisiert gleichzeitig die Rechte

Definiert in Zeile 72 der Datei /impl/GenericHibernateDAOImpl.java.

void setSession (Session s)

Setzt die Hibernate Session der DAO

Parameter:

s zu setzende Session

Definiert in Zeile 95 der Datei /impl/GenericHibernateDAOImpl.java.

1.11.68 AuthDAO Schnittstellenreferenz

Basisklasse für EAMAttributeDAO und GenericDAO< T, ID extends Serializable >.

1.11.68.1 Ausführliche Beschreibung

Interface um Benutzerdaten zur Rechtefilterung für ein DAO zu setzen

Definiert in Zeile 6 der Datei AuthDAO.java.

Öffentliche Methoden

- void reloadRights ()
- void setCredentials (String userid, String password)

1.11.68.2 Dokumentation der Elementfunktionen

void reloadRights ()

Lädt die Rechte für den Benutzer erneut

void setCredentials (String userid, String password)

Erlaubt das setzen von Benutzer Id (i.d.R. ein Long) und Passwort

Parameter:

userid Id des Benutzers

password Passwort des Benutzers (MD5 kodiert)

1.11.69 EAMObjectDAO Schnittstellenreferenz

Abgeleitet von

de::offis::pg::eam::core::metamodel::metainterface::dao::interfaces::GenericDAO< de::offis::pg::eam::core::metamodel::metadata::EAMObject, Long >.

Basisklasse für EAMObjectDAOHibernate.

1.11.69.1 Ausführliche Beschreibung

DAO zum Zugriff auf EAMObject Objekte

Definiert in Zeile 12 der Datei /interfaces/EAMObjectDAO.java.

Öffentliche Methoden

- boolean nameExistsInCategory (String name, EAMCategory cat)

Paketfunktionen

- List< EAMObject > findByCategory (EAMCategory cat)

1.11.69.2 Dokumentation der Elementfunktionen

List<EAMObject> findByCategory (EAMCategory cat) [package]

Rückgabe:

Alle sichtbaren EAMObject eines Metamodells

Implementiert in EAMObjectDAOHibernate.

boolean nameExistsInCategory (String name, EAMCategory cat)

Überprüft ob ein EAMObject in der EAMCategory cat existiert, dessen Attribut "name" dem Parameter "name" entspricht

Parameter:

name Zu überprüfendes "name" Attribut

cat Die EAMCategory in der der name des EAMObjects gesucht werden soll

Rückgabe:

true, wenn eine EAMObject existiert, dessen Attribut "name" dem Parameter "name" in der EAMCategory cat entspricht, false andernfalls

1.11.70 GenericDAO< T, ID extends Serializable > Schnittstellenreferenz

Abgeleitet von

de::offis::pg::eam::core::metamodel::metainterface::dao2::interfaces::AuthDAO.

1.11.70.1 Ausführliche Beschreibung

Generisches DAO Interface, welches übliche CRUD-Methoden zur Verfügung stellt

Parameter:

<T> Typ der DAO

<ID> Typ der Id der durch den Typ der DAO bezeichneten Objekte

Definiert in Zeile 12 der Datei /interfaces/GenericDAO.java.

Öffentliche Methoden

- void delete (T entity) throws Exception
- List< T > findAll ()
- List< T > findByExample (T exampleInstance, String...excludeProperty)
- T save (T entity) throws Exception

Paketfunktionen

- T findById (ID id, boolean lock)

1.11.70.2 Dokumentation der Elementfunktionen

void delete (T entity) throws Exception

Löscht ein persistentes Objekt aus der Datenbank

Parameter:

entity Persistentes Objekt, dass gelöscht werden soll

Ausnahmebehandlung:**Exception****List<T> findAll ()**

Sucht alle Objekte des Typs

Rückgabe:

Liste mit allen Objekten

List<T> findByExample (T exampleInstance, String... excludeProperty)

Sucht Objekte anhand von Beispielinstanzen

Beispiel:

```
EAMCategory category = new EAMCategory();
category.setModelCategory("IT-Infrastruktur");
category.setDescription("Beschreibung der IT-Infrastruktur ...");
...
// sucht nach Kategorien mit dem Namen "IT-Infrastruktur",
//berücksichtigt aber nicht deren Beschreibung
dao.findByExample(category,"description");
```

Parameter:

exampleInstance Beispielinstantz mit entsprechende gesetzten Attributen

excludeProperty Variable Anzahl von Strings, welche Parameter der Beispielinstantz angeben, die bei der Suche nicht berücksichtigt werden sollen

Rückgabe:

Liste mit Objekten die der Beispielinstantz (bzw. mit ausgeschlossenen Parametern) entsprechen

T findById (ID id, boolean lock) [package]

Sucht ein Objekt anhand der Id

Parameter:

id Id (Datenbank-Identität) des Objektes

lock Gibt an, ob das Objekt für Veränderung gesperrt werden soll

Rückgabe:

Gesuchtes Objekt oder Exception falls keines gefunden wird

T save (T entity) throws Exception

Speichert ein Objekt in der Datenbank

Parameter:

entity Objekt welches persistiert oder aktualisiert werden soll

Rückgabe:

Aktualisiertes Objekt mit den Eigenschaften, die möglicherweise durch die Datenbank zugewiesen worden (z.B. Id)

Ausnahmebehandlung:

Exception

1.11.71 DAOFactory Klassenreferenz

Basisklasse für HibernateDAOFactoryImpl.

1.11.71.1 Ausführliche Beschreibung

Factory um DAOs zu erzeugen. Anhand der DAOs kann ein Zugriff auf die gewünschten Entitäten erfolgen.

Die DAOs dürfen nur innerhalb einer Transaktion verwendet werden.

Dies ist durch die Verwendung des Servletfilters HibernateTransactionServletfilter möglich - dort sind auch Methoden, welche eine manuelle Transaktionsdemarkation erlauben.

Definiert in Zeile 28 der Datei factory/DAOFactory.java.

Öffentliche Methoden

- abstract EAMAttributeDAO getEAMAttributeDAO ()
- abstract EAMAttributeTypeDAO getEAMAttributeTypeDAO ()
- abstract EAMCategoryDAO getEAMCategoryDAO ()
- abstract EAMObjectDAO getEAMObjectDAO ()
- abstract EAMObjectTypeDAO getEAMObjectTypeDAO ()
- abstract EAMRelationDAO getEAMRelationDAO ()
- abstract EAMRelationTypeDAO getEAMRelationTypeDAO ()
- abstract EAMStatusDAO getEAMStatusDAO ()
- abstract EAMTimeDAO getEAMTimeDAO ()

- abstract EAMTypeDAO getEAMTypeDAO ()

Öffentliche, statische Methoden

- static DAOFactory instance (Class factory)

Statische öffentliche Attribute

- static final Class HIBERNATE = HibernateDAOFactoryImpl.class

1.11.71.2 Dokumentation der Elementfunktionen

abstract EAMAttributeDAO getEAMAttributeDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMAttribute

abstract EAMAttributeTypeDAO getEAMAttributeTypeDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMAttributeType

abstract EAMCategoryDAO getEAMCategoryDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMCategory

Wird benutzt von MetamodelAuth.getAttributesByMetamodel().

abstract EAMObjectDAO getEAMObjectDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMObject

Wird benutzt von MetamodelAuth.getAttributesByMetamodel().

abstract EAMObjectTypeDAO getEAMObjectTypeDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMObjectType

abstract EAMRelationDAO getEAMRelationDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMRelation

Wird benutzt von MetamodelAuth.getAttributesByMetamodel().

abstract EAMRelationTypeDAO getEAMRelationTypeDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMRelationType

abstract EAMStatusDAO getEAMStatusDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMStatus

abstract EAMTimeDAO getEAMTimeDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMTime

abstract EAMTypeDAO getEAMTypeDAO () [pure virtual]

Rückgabe:

Data Access Object (DAO) zum Auslesen + Manipulieren von EAMType

static DAOFactory instance (Class factory) [static]

Fabrikmethode zur Instanziierung einer Fabrik

Definiert in Zeile 43 der Datei factory/DAOFactory.java.

1.11.71.3 Dokumentation der Datenelemente

final Class HIBERNATE = HibernateDAOFactoryImpl.class [static]

Erstellt eine alleinstehende DAOFactory, die "unmanaged" DAO beans zurückgibt die in jeder Umgebung für die Hibernate konfiguriert wurde benutzt werden kann. Benutzt HibernateUtil/AbstractSessionFactory und Hibernate context propagation (CurrentSessionContext), Thread-gebunden oder Transaktion- gebunden und Transaktions-gültig

Definiert in Zeile 38 der Datei factory/DAOFactory.java.

1.11.72 HibernateDAOFactoryImpl Klassenreferenz

Abgeleitet von

de::offis::pg::eam::core::metamodel::metainterface::dao::factory::DAOFactory.

1.11.72.1 Ausführliche Beschreibung

Implementierung der DAOFactory mit Hibernate

Diese Klasse enthält ferner einige private Klassen, die DAO Implementierungen darstellen und direkt von GenericDAO abgeleitet sind

Siehe auch:

DAOFactory

Definiert in Zeile 38 der Datei factory/HibernateDAOFactoryImpl.java.

1.11.73 GenericDAO< T, ID extends Serializable > Schnittstellenreferenz

1.11.73.1 Ausführliche Beschreibung

Generisches DAO Interface, welches übliche CRUD-Methoden zur Verfügung stellt

Parameter:

<T> Typ der DAO

<ID> Typ der Id der durch den Typ der DAO bezeichneten Objekte

Definiert in Zeile 12 der Datei interfaces/GenericDAO.java.

Öffentliche Methoden

- void delete (T entity) throws Exception
- List< T > findAll ()
- List< T > findByExample (T exampleInstance, String...excludeProperty)
- T save (T entity) throws Exception

Paketfunktionen

- T findById (ID id, boolean lock)

1.11.73.2 Dokumentation der Elementfunktionen

void delete (T entity) throws Exception

Löscht ein persistentes Objekt aus der Datenbank

Parameter:

entity Persistentes Objekt, dass gelöscht werden soll

Ausnahmebehandlung:

Exception

List<T> findAll ()

Sucht alle Objekte des Typs

Rückgabe:

Liste mit allen Objekten

List<T> findByExample (T exampleInstance, String... excludeProperty)

Sucht Objekte anhand von Beispielinstanzen

Beispiel:

```
EAMCategory category = new EAMCategory();
category.setModelCategory("IT-Infrastruktur");
category.setDescription("Beschreibung der IT-Infrastruktur ...");
...
// sucht nach Kategorien mit dem Namen "IT-Infrastruktur",
//berücksichtigt aber nicht deren Beschreibung
dao.findByExample(category,"description");
```

Parameter:

exampleInstance Beispielinstantz mit entsprechende gesetzten Attributen

excludeProperty Variable Anzahl von Strings, welche Parameter der Beispielinstantz angeben, die bei der Suche nicht berücksichtigt werden sollen

Rückgabe:

Liste mit Objekten die der Beispielinstantz (bzw. mit ausgeschlossenen Parametern) entsprechen

T findById (ID id, boolean lock) [package]

Sucht ein Objekt anhand der Id

Parameter:

id Id (Datenbank-Identität) des Objektes

lock Gibt an, ob das Objekt für Veränderung gesperrt werden soll

Rückgabe:

Gesuchtes Objekt oder Exception falls keines gefunden wurde

T save (T entity) throws Exception

Speichert ein Objekt in der Datenbank

Parameter:

entity Objekt welches persistiert oder aktualisiert werden soll

Rückgabe:

Aktualisiertes Objekt mit den Eigenschaften, die möglicherweise durch die Datenbank zugewiesen worden (z.B. Id)

Ausnahmebehandlung:

Exception

1.11.74 MetaInterface Klassenreferenz**1.11.74.1 Ausführliche Beschreibung****Siehe auch:**

DAOFactory, DAOFactory#getInstance()

Definiert in Zeile 24 der Datei MetaInterface.java.

Öffentliche Methoden

- boolean createObject (Object obj)
- boolean deleteObject (Object obj)
- List< EAMObject > findAllEAMObjects (EAMType type)
- List< EAMRelation > findAllEAMRelations (EAMType type)
- EAMObject findEAMObjectById (long id)
- EAMRelation findEAMRelationById (long id)
- Object updateObject (Object obj)

1.11.74.2 Dokumentation der Elementfunktionen**boolean createObject (Object obj)**

Methode zu Anlegen von EAM Objekten und EAM Relationen

Parameter:

obj

Rückgabe:

true falls das Anlegen erfolgreich war, false andernfalls

Definiert in Zeile 42 der Datei MetaInterface.java.

boolean deleteObject (Object obj)

Entfernt persistente Objekt aus der Datenbank

Parameter:

obj persistentes Objekt welches gelöscht werden soll

Rückgabe:

true falls das Objekt gelöscht wurde, false andernfalls

Definiert in Zeile 89 der Datei MetaInterface.java.

List<EAMObject> findAllEAMObjects (EAMType type)

Lädt alle EAMObjekte eines bestimmten Typs

Parameter:

type Typ anhand dessen alle zugehörigen EAM Objekte geladen werden sollen

Rückgabe:

Liste mit EAM Objekten die von dem Typ sind

Definiert in Zeile 124 der Datei MetaInterface.java.

List<EAMRelation> findAllEAMRelations (EAMType type)

Lädt alle EAMObjekte eines bestimmten Typs

Parameter:

type Typ anhand dessen alle zugehörigen EAM Objekte geladen werden sollen

Rückgabe:

Liste mit EAM Objekten die von dem Typ sind

Definiert in Zeile 147 der Datei MetaInterface.java.

EAMObject findEAMObjectById (long id)

Lädt ein EAMObject anhand des Namens

Parameter:

name Name der zur bestimmung des EAMObjects dient

Rückgabe:

EAMObject das dem angegebenen Kriterium (name) entspricht

Definiert in Zeile 112 der Datei MetaInterface.java.

EAMRelation findEAMRelationById (long id)

Lädt ein EAMObject anhand des Namens

Parameter:

name Name der zur bestimmung des EAMObjects dient

Rückgabe:

EAMObject das dem angegebenen Kriterium (name) entspricht

Definiert in Zeile 135 der Datei MetaInterface.java.

Object updateObject (Object obj)

Diese Methode erlaubt es eine EAMObject oder eine EAMRelation zu aktualisieren (update)

Parameter:

obj persistentes Objekt welches aktualisiert werden soll

Rückgabe:

Das aktualisierte Objekt, null andernfalls

Definiert in Zeile 66 der Datei MetaInterface.java.

1.11.75 MetamodelHelper Klassenreferenz**1.11.75.1 Ausführliche Beschreibung****Autor:**

Roland Koppe

Definiert in Zeile 32 der Datei MetamodelHelper.java.

Öffentliche, statische Methoden

- static String makeUniqueName (Object object, String postfix)

1.11.75.2 Dokumentation der Elementfunktionen

static String makeUniqueName (Object object, String postfix) [static]

Prüft den durch das angegebene MetaObjekt angegebenen Namen auf Eindeutigkeit in der Datenbank. Sollte der angegebene Name schon vergeben sein, so wird der angegebene Postfix angefügt. Sollte dieser immer noch nicht eindeutig sein, wird eine Nummerierung mit 1 beginnend angefügt. Der eindeutige Name wird dann zurückgeliefert.

Parameter:

object Object Metamodell Object

postfix String Postfix

Rückgabe:

String eindeutiger Name, null, wenn nicht erfolgreich oder DB Exception

Definiert in Zeile 46 der Datei MetamodelHelper.java.

1.11.76 IQueryDAO Schnittstellenreferenz

1.11.76.1 Ausführliche Beschreibung

AUTO-CREATED INTERFACE FOR
de.offis.pg.eam.core.metamodel.query.api.QueryDAO

Definiert in Zeile 17 der Datei IQueryDAO.java.

1.11.77 QueryDAO Klassenreferenz

1.11.77.1 Ausführliche Beschreibung

DAO für die Speicherung, Anfrage und Ausführung von Queries.

Autor:

Roland Koppe

Definiert in Zeile 29 der Datei QueryDAO.java.

Öffentliche, statische Methoden

- static DataObjectList _executeQuery (Query query, String userId, String password) throws InstanceException, DatabaseException
- static LinkedList< DataObject > _executeQuery (long queryId, String userId, String password) throws InstanceException, DatabaseException
- static boolean deleteQuery (long queryId, String userId, String password)
- static boolean deleteQuery (Query query, String userId, String password)
- static LinkedList< DataObject > executeQuery (Query query, String userId, String password) throws InstanceException, DatabaseException

- static LinkedList< DataObject > executeQuery (long queryId, String userId, String password) throws InstanceException, DatabaseException
- static LinkedList< Query > loadQueries (long id, String userId, String password)
- static long saveQuery (Query query, String userId, String password) throws DatabaseException

1.11.77.2 Dokumentation der Elementfunktionen

static DataObjectList _executeQuery (Query query, String userId, String password) throws InstanceException, DatabaseException [static]

Führt die angegebene Query aus und liefert das Ergebnis als DataObjectList.

Parameter:

query Query

userId String ID des Benutzers

password String Passwort des Benutzers

Rückgabe:

DataObjectList

Ausnahmebehandlung:

InstanceException

DatabaseException

Definiert in Zeile 275 der Datei QueryDAO.java.

static LinkedList<DataObject> _executeQuery (long queryId, String userId, String password) throws InstanceException, DatabaseException [static]

Führt die angegebene Query aus und liefert das Ergebnis als DataObjectList.

Parameter:

queryId long ID der Query

userId String ID des Benutzers

password String Passwort des Benutzers

Rückgabe:

DataObjectList

Ausnahmebehandlung:

InstanceException

DatabaseException

Definiert in Zeile 253 der Datei QueryDAO.java.

Benutzt QueryDAO.executeQuery() und QueryDAO.loadQueries().

static boolean deleteQuery (long queryId, String userId, String password) [static]

Löscht die angegebene Query mit den angegebenen Benutzerdaten.

Parameter:

long ID des Query

userId String ID des Benutzers

password String Passwort des Benutzers

Rückgabe:

boolean true, wenn erfolgreich

Definiert in Zeile 113 der Datei QueryDAO.java.

static boolean deleteQuery (Query query, String userId, String password) [static]

Löscht die angegebene Query mit den angegebenen Benutzerdaten.

Parameter:

query Query

userId String ID des Benutzers

password String Passwort des Benutzers

Rückgabe:

boolean true, wenn erfolgreich

Definiert in Zeile 95 der Datei QueryDAO.java.

Benutzt Query.getId().

static LinkedList<DataObject> executeQuery (Query query, String userId, String password) throws InstanceException, DatabaseException [static]

Führt die angegebene Query aus und liefert das Ergebnis als Liste von DataObjects.

Parameter:

query Query

userId String ID des Benutzers

password String Passwort des Benutzers

Rückgabe:

LinkedList<DataObject> Liste von DataObjects

Ausnahmebehandlung:

InstanceException

DatabaseException

Definiert in Zeile 230 der Datei QueryDAO.java.

static LinkedList<DataObject> executeQuery (long queryId, String userId, String password) throws InstanceException, DatabaseException [static]

Führt die angegebene Query aus und liefert das Ergebnis als Liste von DataObjects.

Parameter:

queryId long ID der Query

userId String ID des Benutzers

password String Passwort des Benutzers

Rückgabe:

LinkedList<DataObject> Liste von DataObjects

Ausnahmebehandlung:

InstanceException

DatabaseException

Definiert in Zeile 208 der Datei QueryDAO.java.

Benutzt QueryDAO.loadQueries().

Wird benutzt von QueryDAO._executeQuery().

static LinkedList<Query> loadQueries (long id, String userId, String password) [static]

Holt alle Queries, die von dem angegebenen Benutzer gesehen werden können. Ist das Argument id mit 0 angegeben, so werden alle Queries geliefert. Wird eine id angegeben, so wird die entsprechende Query geliefert.

Parameter:

id long optional, ID der Query

userId String ID des Benutzers

password String Passwort des Benutzers

Rückgabe:

LinkedList<Query> Liste von Queries

Definiert in Zeile 142 der Datei QueryDAO.java.

Benutzt Query.setCreator(), Query.setDate(), Query.setGlobal(), Query.setGlobalEditable() und Query.setId().

Wird benutzt von QueryDAO._executeQuery() und QueryDAO.executeQuery().

static long saveQuery (Query query, String userId, String password) throws DatabaseException [static]

Speichert die angegebene Query unter den angegebenen Benutzerdaten. Hat die angegebene Query eine ID != 0, so wird sie hinzugefügt, sonst geändert.

Parameter:

query Query
userId String ID des Benutzers
password String Passwort der Benutzers

Rückgabe:

long ID der gespeicherten Query

Ausnahmebehandlung:

DatabaseException

Definiert in Zeile 43 der Datei QueryDAO.java.

1.11.78 Query Klassenreferenz

1.11.78.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 15 der Datei query/model/Query.java.

1.12 module

1.12.1 IModuleView Schnittstellenreferenz

Basisklasse für BundleViews.

1.12.1.1 Ausführliche Beschreibung

Dieses Interface wird dazu benutzt, um ein Bundle am Kern anzumelden. Uebergeben werden der Name des Bundles und eine Beschreibung sowie eine Liste von Rollen, die dieses Bundle definiert.

Autor:

Yu, Christian Z

Definiert in Zeile 13 der Datei IModuleView.java.

1.13 property

1.13.1 EAMProperties Klassenreferenz

1.13.1.1 Ausführliche Beschreibung

Properties des EAM-Tools. Die Eigenschaften des EAM-Tools werden bei der ersten Verwendung vollständig geladen, d.h. ein manuelles Laden ist nicht erforderlich

Autor:

Roland Koppe

Definiert in Zeile 30 der Datei EAMProperties.java.

Öffentliche, statische Methoden

- static Properties getProperties ()
- static String getProperty (EAMProperty prop)
- static String getProperty (String key)
- static synchronized void load ()
- static void setProperty (EAMProperty prop, String value)
- static void setProperty (String key, String value)
- static void setPropertyDefault (EAMProperty prop, String value)

1.13.1.2 Dokumentation der Elementfunktionen

static Properties getProperties () [static]

Speichert die Properties. Liefert das Properties Objekt.

Rückgabe:

Properties

Definiert in Zeile 243 der Datei EAMProperties.java.

Benutzt EAMProperties.load().

static String getProperty (EAMProperty prop) [static]

Liefert den Wert zum angegebenen Schlüssel.

Parameter:

prop EAMProperty Enum-Wert

Rückgabe:

String Wert

Definiert in Zeile 271 der Datei EAMProperties.java.

Benutzt EAMProperties.load().

static String getProperty (String key) [static]

Liefert den Wert zum angegebenen Schlüssel.

Parameter:

key String Schlüssel, Name des Property

Rückgabe:

String Wert

Definiert in Zeile 257 der Datei EAMProperties.java.

Benutzt EAMProperties.load().

Wird benutzt von EAMProperties.load().

static synchronized void load () [static]

Lädt die Properties Datei und prüft, ob alle Properties gesetzt wurden, wenn nicht, werden die Standardwerte verwendet. Standardwerte sind definiert als der Name des Eintrags in der Enum Klasse.

Die Datei wird gesucht als: System.getProperty("user.dir") + "/eamconfig/eam.properties"

Definiert in Zeile 155 der Datei EAMProperties.java.

Benutzt EAMProperties.getProperty().

Wird benutzt von EAMProperties.getProperties() und EAMProperties.getProperty().

static void setProperty (EAMProperty prop, String value) [static]

Setzt den Wert des angegeben Schlüssels.

Parameter:

prop EAMProperty Schlüssel

value String Wert

Definiert in Zeile 298 der Datei EAMProperties.java.

static void setProperty (String key, String value) [static]

Setzt den Wert des angegebenen Schlüssels.

Parameter:

key String Schlüssel

value String Wert

Definiert in Zeile 286 der Datei EAMProperties.java.

static void setPropertyDefault (EAMProperty prop, String value) [static]

Setzt den Standard-Wert zum angegebenen Schlüssel.

Parameter:

prop EAMProperty Schlüssel

value String Wert

Definiert in Zeile 310 der Datei EAMProperties.java.

1.14 proxy

1.14.1 EAMProxy Klassenreferenz

Abgeleitet von java::lang::reflect::InvocationHandler.

1.14.1.1 Ausführliche Beschreibung

Diese Klasse ermöglicht das Aufrufen von statischen Methoden aus dem Kern, die durch ein Interface beschrieben sind.

Autor:

Roland Koppe

David Heimann

Definiert in Zeile 28 der Datei EAMProxy.java.

Öffentliche Methoden

- Object invoke (Object object, Method method, Object[] args) throws Throwable

Öffentliche, statische Methoden

- static Object newInstance (Class interfaze, long userId, String password) throws ProxyException, Throwable

1.14.1.2 Dokumentation der Elementfunktionen

Object invoke (Object object, Method method, Object[] args) throws Throwable

Ruft die angegebene Methode des Objects mit den benannten Argumenten auf und liefert das Ergebnis als Object zurück.

Parameter:

object Object
method Method
args Object[]

Rückgabe:

Object

Ausnahmebehandlung:

ProxyInvocationException

Definiert in Zeile 114 der Datei EAMProxy.java.

static Object newInstance (Class interfaze, long userId, String password) throws ProxyException, Throwable [static]

Erzeugt eine neue Instanz eines Proxys auf die Implementierung des angegebenen Interfaces.

Parameter:

interfaze Class Interface

Rückgabe:

Object als Referenz der Implementierung

Ausnahmebehandlung:

ProxyImplementationClassException

Definiert in Zeile 86 der Datei EAMProxy.java.

1.14.2 Crawler Klassenreferenz

1.14.2.1 Ausführliche Beschreibung

Durchsucht den Kern mit Hilfe der Manifest Definition nach öffentlichen Klassen und Methoden.

Autor:

Roland Koppe

Definiert in Zeile 21 der Datei Crawler.java.

Öffentliche Methoden

- Crawler (String manifestFile)
- ArrayList< String > getPublicClasses ()
- ArrayList< String > getPublicMethods ()
- ArrayList< String > getPublicPackages ()

1.14.2.2 Beschreibung der Konstruktoren und Destruktoren

Crawler (String manifestFile)

Konstruktor.

Parameter:

manifestFile String Dateiname des Manifests.

Definiert in Zeile 34 der Datei Crawler.java.

1.14.2.3 Dokumentation der Elementfunktionen

ArrayList<String> getPublicClasses ()

Liefert die öffentlichen Klassen des Manifests.

Rückgabe:

ArrayList<String>

Definiert in Zeile 60 der Datei Crawler.java.

ArrayList<String> getPublicMethods ()

Liefert die öffentlichen Methoden des Manifests.

Rückgabe:

ArrayList<String>

Definiert in Zeile 71 der Datei Crawler.java.

ArrayList<String> getPublicPackages ()

Liefert die öffentlichen Pakete des Manifests.

Rückgabe:

ArrayList<String>

Definiert in Zeile 49 der Datei Crawler.java.

1.14.3 CrawlerCore Klassenreferenz

1.14.3.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei CrawlerCore.java.

1.14.4 MethodCrawler Klassenreferenz

1.14.4.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei MethodCrawler.java.

1.14.5 MethodDescription Klassenreferenz

1.14.5.1 Ausführliche Beschreibung

Diese Klasse definiert eine Methodenbeschreibung, die Namen und Javadoc einer Methode enthält. Außerdem wird der Klassenname und der Name des Pakets gespeichert.

Autor:

Roland Koppe

Definiert in Zeile 13 der Datei MethodDescription.java.

1.14.6 MethodDescriptionCrawler Klassenreferenz

1.14.6.1 Ausführliche Beschreibung

Durchsucht das angegebene Manifest nach exportierten Klassen und liefert deren statische und öffentliche Methoden mit Javadoc.

Autor:

Roland Koppe

Definiert in Zeile 21 der Datei MethodDescriptionCrawler.java.

Öffentliche Methoden

- `ArrayList< MethodDescription > getMethodDescriptions ()`
- `MethodDescriptionCrawler (String manifestFile)`

1.14.6.2 Beschreibung der Konstruktoren und Destruktoren

MethodDescriptionCrawler (String manifestFile)

Durchsucht das angegebene Manifest nach exportierten Klassen und liefert deren statische und öffentliche Methoden mit Javadoc.

Parameter:

manifestFile String Dateiname des Manifests

Definiert in Zeile 34 der Datei MethodDescriptionCrawler.java.

1.14.6.3 Dokumentation der Elementfunktionen

ArrayList<MethodDescription> getMethodDescriptions ()

Liefert die gefundenen Methoden als ArrayList.

Rückgabe:

ArrayList<MethodDescription>

Definiert in Zeile 51 der Datei MethodDescriptionCrawler.java.

1.14.7 Crawler Klassenreferenz

1.14.7.1 Ausführliche Beschreibung

Durchsucht den Kern mit Hilfe der Manifest Definition nach öffentlichen Klassen und Methoden.

Autor:

Roland Koppe

Definiert in Zeile 21 der Datei sql/Crawler.java.

Öffentliche Methoden

- Crawler (String manifestFile)
- ArrayList< String > getPublicClasses ()
- ArrayList< String > getPublicMethods ()
- ArrayList< String > getPublicPackages ()

1.14.7.2 Beschreibung der Konstruktoren und Destruktoren

Crawler (String manifestFile)

Konstruktor.

Parameter:

manifestFile String Dateiname des Manifests.

Definiert in Zeile 34 der Datei sql/Crawler.java.

1.14.7.3 Dokumentation der Elementfunktionen

ArrayList<String> getPublicClasses ()

Liefert die öffentlichen Klassen des Manifests.

Rückgabe:

ArrayList<String>

Definiert in Zeile 60 der Datei sql/Crawler.java.

ArrayList<String> getPublicMethods ()

Liefert die öffentlichen Methoden des Manifests.

Rückgabe:

ArrayList<String>

Definiert in Zeile 71 der Datei sql/Crawler.java.

ArrayList<String> getPublicPackages ()

Liefert die öffentlichen Pakete des Manifests.

Rückgabe:

ArrayList<String>

Definiert in Zeile 49 der Datei sql/Crawler.java.

1.14.8 CrawlerCore Klassenreferenz

1.14.8.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei sql/CrawlerCore.java.

1.14.9 MethodCrawler Klassenreferenz

1.14.9.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei sql/MethodCrawler.java.

1.14.10 MethodDescription Klassenreferenz

1.14.10.1 Ausführliche Beschreibung

Diese Klasse definiert eine Methodenbeschreibung, die Namen und Javadoc einer Methode enthält. Außerdem wird der Klassenname und der Name des Pakets gespeichert.

Autor:

Roland Koppe

Definiert in Zeile 13 der Datei sql/MethodDescription.java.

1.14.11 MethodDescriptionCrawler Klassenreferenz

1.14.11.1 Ausführliche Beschreibung

Durchsucht das angegebene Manifest nach exportierten Klassen und liefert deren statische und öffentliche Methoden mit Javadoc.

Autor:

Roland Koppe

Definiert in Zeile 19 der Datei sql/MethodDescriptionCrawler.java.

Öffentliche Methoden

- `ArrayList< MethodDescription > getMethodDescriptions ()`
- `MethodDescriptionCrawler (String manifestFile)`

1.14.11.2 Beschreibung der Konstruktoren und Destruktoren

MethodDescriptionCrawler (String manifestFile)

Durchsucht das angegebene Manifest nach exportierten Klassen und liefert deren statische und öffentliche Methoden mit Javadoc.

Parameter:

manifestFile String Dateiname des Manifests

Definiert in Zeile 32 der Datei sql/MethodDescriptionCrawler.java.

1.14.11.3 Dokumentation der Elementfunktionen

ArrayList<MethodDescription> getMethodDescriptions ()

Liefert die gefundenen Methoden als ArrayList.

Rückgabe:

ArrayList<MethodDescription>

Definiert in Zeile 49 der Datei sql/MethodDescriptionCrawler.java.

1.14.12 ProxyException Klassenreferenz

Basisklasse für ProxyImplementationClassException, ProxyInterfaceException, ProxyInvocationException und ProxyInvocationUserNotExistsException.

1.14.12.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei ProxyException.java.

1.14.13 ProxyImplementationClassException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::proxy::exceptions::ProxyException.

1.14.13.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei ProxyImplementationClassException.java.

1.14.14 ProxyInterfaceException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::proxy::exceptions::ProxyException.

1.14.14.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei ProxyInterfaceException.java.

1.14.15 ProxyInvocationException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::proxy::exceptions::ProxyException.

Basisklasse für ProxyInvocationUserHasNoRightsException.

1.14.15.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei ProxyInvocationException.java.

1.14.16 ProxyInvocationUserHasNoRightsException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::proxy::exceptions::ProxyInvocationException.

1.14.16.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei ProxyInvocationUserHasNoRightsException.java.

1.14.17 ProxyInvocationUserNotExistsException Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::proxy::exceptions::ProxyException.

1.14.17.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei ProxyInvocationUserNotExistsException.java.

1.14.18 ProxyInterface Schnittstellenreferenz

1.14.18.1 Ausführliche Beschreibung

Annotation für Implementierungsklassen, die über ein Interface nach außen gegeben werden sollen.

Autor:

Roland Koppe

Definiert in Zeile 20 der Datei ProxyInterface.java.

1.14.19 ProxyInterfaceBuilder Klassenreferenz

1.14.19.1 Ausführliche Beschreibung

Diese Klasse baut aus der angegebenen Klasse ein Interface.

Autor:

Roland Koppe

Definiert in Zeile 22 der Datei ProxyInterfaceBuilder.java.

Öffentliche, statische Methoden

- static boolean buildInterface (Class clazz) throws ProxyException

1.14.19.2 Dokumentation der Elementfunktionen

static boolean buildInterface (Class clazz) throws ProxyException [static]

Baut aus der angegebenen Klasse ein Interface. Dabei werden nur statische Methoden betrachtet. Die angegebene Klasse muss mit ProxyInterface annotiert sein und die Annotation muss

Parameter:

clazz

Rückgabe:

Ausnahmebehandlung:

ProxyException

Definiert in Zeile 34 der Datei ProxyInterfaceBuilder.java.

1.14.20 IAuthUser Schnittstellenreferenz

1.14.20.1 Ausführliche Beschreibung

AUTO-CREATED INTERFACE FOR de.offis.pg.eam.core.invoke.test.api.AuthUser

Definiert in Zeile 10 der Datei IAuthUser.java.

1.15 test

1.15.1 LockThread Klassenreferenz

1.15.1.1 Ausführliche Beschreibung

Threads sollen Locking "simulieren".

Autor:

Roland Koppe

Definiert in Zeile 19 der Datei LockThread.java.

1.15.2 Test_AuthNames Klassenreferenz

Abgeleitet von junit::framework::TestCase.

1.15.2.1 Ausführliche Beschreibung

Autor:

roko

Definiert in Zeile 16 der Datei Test_AuthNames.java.

1.15.3 Test_Database Klassenreferenz

1.15.3.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 14 der Datei Test_Database.java.

1.15.4 Test_DatabaseInsert Klassenreferenz

Abgeleitet von junit::framework::TestCase.

1.15.4.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 16 der Datei Test_DatabaseInsert.java.

1.15.5 Test_DataObject Klassenreferenz

Abgeleitet von junit::framework::TestCase.

1.15.5.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 22 der Datei Test_DataObject.java.

1.15.6 Test_DataObjectList Klassenreferenz

1.15.6.1 Ausführliche Beschreibung

Test der Anfrage von DataObjectList

Autor:

Roland Koppe

Definiert in Zeile 24 der Datei Test_DataObjectList.java.

1.15.7 Test_DataTypeValidator Klassenreferenz

Abgeleitet von junit::framework::TestCase.

1.15.7.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 18 der Datei Test_DataTypeValidator.java.

1.15.8 Test_DOInsertVerantwortlicher Klassenreferenz

1.15.8.1 Ausführliche Beschreibung

Autor:

roko

Definiert in Zeile 13 der Datei Test_DOInsertVerantwortlicher.java.

1.15.9 Test_EAMProperties Klassenreferenz

Abgeleitet von junit::framework::TestCase.

1.15.9.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 16 der Datei Test_EAMProperties.java.

1.15.10 Test_EAMProxy Klassenreferenz

1.15.10.1 Ausführliche Beschreibung

Autor:

roko

Definiert in Zeile 18 der Datei Test_EAMProxy.java.

1.15.11 Test_Hibernate Klassenreferenz

1.15.11.1 Ausführliche Beschreibung

Autor:

roko

Definiert in Zeile 34 der Datei Test_Hibernate.java.

1.15.12 Test_ImportExport Klassenreferenz

1.15.12.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei Test_ImportExport.java.

1.15.13 Test_Instance_Refactor1 Klassenreferenz

1.15.13.1 Ausführliche Beschreibung

Test von Instance und Query nach Änderung der DataObject hashMap von Attributnamen nach ids.

Autor:

Roland Koppe

Definiert in Zeile 25 der Datei Test_Instance_Refactor1.java.

1.15.14 Test_InstanceDataInterface Klassenreferenz

Abgeleitet von junit::framework::TestCase.

1.15.14.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 23 der Datei Test_InstanceDataInterface.java.

1.15.15 Test_InstanceDelete Klassenreferenz

1.15.15.1 Ausführliche Beschreibung

Test von durch Relationen verbundene Instanzen. Das Löschen einer Objekt-Instanz führt zum Löschen aller Relation-Instanzen an denen die Objekt-Instanz beteiligt war.

Autor:

Roland Koppe

Definiert in Zeile 21 der Datei Test_InstanceDelete.java.

1.15.16 Test_InstanceInterface Klassenreferenz

1.15.16.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei Test_InstanceInterface.java.

1.15.17 Test_InstanceLocking Klassenreferenz

Abgeleitet von junit::framework::TestCase.

1.15.17.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 19 der Datei Test_InstanceLocking.java.

1.15.18 Test_InstanceLockingThread Klassenreferenz

1.15.18.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 16 der Datei Test_InstanceLockingThread.java.

1.15.19 Test_InstanceQuery Klassenreferenz

Abgeleitet von junit::framework::TestCase.

1.15.19.1 Ausführliche Beschreibung

Sammlung von Testfällen für Query.

Autor:

Roland Koppe

Definiert in Zeile 22 der Datei Test_InstanceQuery.java.

1.15.20 Test_LoggingFiles Klassenreferenz

1.15.20.1 Ausführliche Beschreibung

Autor:

roko

Definiert in Zeile 13 der Datei Test_LoggingFiles.java.

1.15.21 Test_MetamodelAuth Klassenreferenz

1.15.21.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 15 der Datei Test_MetamodelAuth.java.

1.15.22 Test_MetamodelHelper Klassenreferenz

1.15.22.1 Ausführliche Beschreibung

Autor:

roko

Definiert in Zeile 13 der Datei Test_MetamodelHelper.java.

1.15.23 Test_SQL_Split Klassenreferenz

1.15.23.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 15 der Datei Test_SQL_Split.java.

1.15.24 Test_SQLEscape Klassenreferenz

Abgeleitet von junit::framework::TestCase.

1.15.24.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 13 der Datei Test_SQLEscape.java.

1.15.25 Test_XMLValidator Klassenreferenz

Abgeleitet von junit::framework::TestCase.

1.15.25.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 13 der Datei Test_XMLValidator.java.

1.16 tools

1.16.1 File Klassenreferenz

1.16.1.1 Ausführliche Beschreibung

Die Klasse erlaubt das einfache Lesen und Schreiben von Dateien.

Autor:

Roland Koppe

Definiert in Zeile 19 der Datei File.java.

Öffentliche, statische Methoden

- static List<String> findFiles (String directory, String prefix, String postfix)
- static String read (String filename)
- static boolean write (String filename, boolean append, String content)

1.16.1.2 Dokumentation der Elementfunktionen

static List<String> findFiles (String directory, String prefix, String postfix)
[static]

Liefert alle Dateien eines Verzeichnisses inkl. Unterverzeichnisse, die im angegebenen Verzeichnis den angegebenen Präfix und Postfix erfüllen.

Parameter:

directory String Verzeichnis
prefix String Präfix der Datei
postfix String Postfix der Datei

Rückgabe:

List<String> Liste von Dateien und Verzeichnissen

Definiert in Zeile 111 der Datei File.java.

static String read (String filename) [static]

Liest die angegebene Datei ein. Sollte während des Lesens ein Fehler auftreten wird null zurückgeliefert. Sonst liefert die Methode den Inhalt der Datei als String.

Parameter:

filename String Dateiname

Rückgabe:

String Inhalt

Definiert in Zeile 30 der Datei File.java.

static boolean write (String filename, boolean append, String content) [static]

Schreibt den angegebenen Inhalt in die angegebene Datei. Ist der Flag append gesetzt, so wird die Inhalt angefügt, sonst eine bestehende Datei überschrieben.

Parameter:

filename String Dateiname
append boolean Inhalt anfügen

content String Inhalt

Rückgabe:

true, wenn erfolgreich.

Definiert in Zeile 71 der Datei File.java.

1.16.2 SQLTools Klassenreferenz

1.16.2.1 Ausführliche Beschreibung

Diese Klasse enthält Werkzeuge für SQL.

Autor:

Roland Koppe

Definiert in Zeile 11 der Datei SQLTools.java.

Öffentliche, statische Methoden

- static String escape (String string)

1.16.2.2 Dokumentation der Elementfunktionen

static String escape (String string) [static]

Die angegebene Zeichenkette wird escaped, d.h. Anführungszeichen und Backslashes werden aus- geklammert.

Parameter:

string String

Rückgabe:

String escaped

Definiert in Zeile 22 der Datei SQLTools.java.

1.16.3 MessageTag Klassenreferenz

Abgeleitet von javax::servlet::jsp::tagext::TagSupport.

1.16.3.1 Ausführliche Beschreibung

Diese Klasse ist fuer die Taglibrary des Sicherheitssystem. Mit Hilfe dieses Tags werden Bereiche auf einer Seite mit einer Rolle/ID/Recht gekennzeichnet. Diese/s Rolle/ID/Recht wird spaeter von Admin dann auf Rollen gemappt die existieren.

Autor:

christian zillmann

Definiert in Zeile 22 der Datei MessageTag.java.

Öffentliche Methoden

- `int doStartTag ()` throws `JspException`

1.16.3.2 Dokumentation der Elementfunktionen**`int doStartTag ()` throws `JspException`**

Definiert in Zeile 63 der Datei MessageTag.java.

1.16.4 Text Klassenreferenz**1.16.4.1 Ausführliche Beschreibung**

Diese Klasse bietet Methoden zum Bearbeiten von Texten.

Autor:

Roland Koppe

Definiert in Zeile 11 der Datei Text.java.

Öffentliche, statische Methoden

- `static String capatalize (String string)`
- `static String repeat (String string, int count)`

1.16.4.2 Dokumentation der Elementfunktionen**`static String capatalize (String string)` [static]**

Liefert den angegeben String als kapatilisierten String wieder. Also "abc hallo welt" nach "abc Hallo Welt".

Parameter:

string String

Rückgabe:

String

Definiert in Zeile 21 der Datei Text.java.

static String repeat (String string, int count) [static]

Der angegebene Text wird entsprechend der Angabe wiederholt und zurückgegeben.

Parameter:

string String

count int Anzahl der Wiederholungen

Rückgabe:

String wiederholter String

Definiert in Zeile 49 der Datei Text.java.

1.16.5 ErrorHandler Klassenreferenz

1.16.5.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 13 der Datei tools/xml/validator/ErrorHandler.java.

1.16.6 Validator Klassenreferenz

1.16.6.1 Ausführliche Beschreibung

XML Validator.

Autor:

Roland Koppe

Definiert in Zeile 35 der Datei tools/xml/validator/Validator.java.

Öffentliche, statische Methoden

- static boolean validateByDOM (String xmlSchemaURL, String xmlDocumentURL)
- static boolean validateBySAX (InputStream xmlSchema, String xmlDocumentURL)
- static boolean validateBySAX (String xmlSchemaURL, String xmlDocumentURL)

1.16.6.2 Dokumentation der Elementfunktionen

static boolean validateByDOM (String xmlSchemaURL, String xmlDocumentURL)
[static]

Validiert das angegebene XML Dokument gegen das angegebene XML Schema und liefert bei Erfolg true zurück. Es wird das DOM benutzt.

Parameter:

xmlSchemaURL URL des XML Schema

xmlDocumentURL URL des XML Dokuments

Rückgabe:

true, wenn Validierung erfolgreich, sonst false

Definiert in Zeile 119 der Datei tools/xml/validator/Validator.java.

static boolean validateBySAX (InputStream xmlSchema, String xmlDocumentURL)
[static]

Validiert das angegebene XML Dokument gegen das angegebene XML Schema und liefert bei Erfolg true zurück. Es wird SAX verwendet.

Parameter:

xmlSchema InputStream des Schema

xmlDocumentURL URL des XML Dokuments

Rückgabe:

true, wenn Validierung erfolgreich, sonst false

Definiert in Zeile 69 der Datei tools/xml/validator/Validator.java.

static boolean validateBySAX (String xmlSchemaURL, String xmlDocumentURL)
[static]

Validiert das angegebene XML Dokument gegen das angegebene XML Schema und liefert bei Erfolg true zurück. Es wird SAX verwendet.

Parameter:

xmlSchemaURL URL des XML Schema

xmlDocumentURL URL des XML Dokuments

Rückgabe:

true, wenn Validierung erfolgreich, sonst false

Definiert in Zeile 47 der Datei tools/xml/validator/Validator.java.

1.16.7 XMLDefinition Klassenreferenz

1.16.7.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei XMLDefinition.java.

1.16.8 XMLTools Klassenreferenz

1.16.8.1 Ausführliche Beschreibung

Diese Klasse bietet Hilfsmethoden für XML-Dokumente.

Autor:

Roland Koppe

Definiert in Zeile 27 der Datei XMLTools.java.

Öffentliche, statische Methoden

- static String date2xml (Date date)
- static Date xml2date (String str) throws ParseException
- static String xmlDocument2string (Document doc)

1.16.8.2 Dokumentation der Elementfunktionen

static String date2xml (Date date) [static]

Wandelt das angegebene Datum in ein XML-Datum um. D.h. das Datum wird im Format yyyy-MM-ddTHH:mm:ss.S zurück geliefert.

Parameter:

date Date

Rückgabe:

String Datum

Definiert in Zeile 71 der Datei XMLTools.java.

static Date xml2date (String str) throws ParseException [static]

Wandelt das angegebene XML-Datum in ein Date-Objekt um und liefert dieses zurück.

Parameter:

str String Datum im Format yyyy-MM-ddTHH:mm:ss.S

Ausnahmebehandlung:

ParseException, wenn das angegebene Datum nicht geparkt werden kann

Rückgabe:

Date Datum

Definiert in Zeile 85 der Datei XMLTools.java.

static String xmlDocument2string (Document doc) [static]

Wandelt das angegebene XML-Dokument in einen einfachen String um und liefert diesen dann zurück.

Parameter:

doc Document

Rückgabe:

String

Definiert in Zeile 37 der Datei XMLTools.java.

1.16.9 Zip Klassenreferenz

1.16.9.1 Ausführliche Beschreibung

Diese Klasse ermöglicht das Zippen von Dateien.

Autor:

Roland Koppe

Definiert in Zeile 21 der Datei Zip.java.

Öffentliche, statische Methoden

- static boolean zip (String destination, HashMap< String, String > files)

1.16.9.2 Dokumentation der Elementfunktionen

static boolean zip (String destination, HashMap< String, String > files) [static]

Zippt die in files als Name, Dateiname angegebenen Dateien und speichert die Zip-Datei unter dem angegebenen Pfad.

Parameter:

destination String Zip-Dateiname

files HashMap<String, String> Liste von Dateien mit Name, Dateiname

Rückgabe:

boolean true, wenn erfolgreich

Definiert in Zeile 33 der Datei Zip.java.

2 de.offis.pg.eam.bundlemanager

2.1 BundleConfiguration

2.1.1 BundleViews Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::module::api::IModuleView.

2.1.1.1 Ausführliche Beschreibung

Definiert in Zeile 20 der Datei BundleViews.java.

Öffentliche Methoden

- String getBundleVersion ()
- String getDescription ()
- long getEquinoxBundleId ()
- String getName ()
- List< View > getViews ()

Statische private Attribute

- static final String VIEWS = "bundle_views.xml"

2.1.1.2 Dokumentation der Elementfunktionen

String getBundleVersion ()

Definiert in Zeile 82 der Datei BundleViews.java.

String getDescription ()

Definiert in Zeile 27 der Datei BundleViews.java.

long getEquinoxBundleId ()

Definiert in Zeile 33 der Datei BundleViews.java.

String getName ()

Definiert in Zeile 39 der Datei BundleViews.java.

List<View> getViews ()

Definiert in Zeile 45 der Datei BundleViews.java.

Benutzt BundleViews.VIEWS.

2.1.1.3 Dokumentation der Datenelemente

final String VIEWS = "bundle_views.xml" [static, private]

Definiert in Zeile 23 der Datei BundleViews.java.

Wird benutzt von BundleViews.getViews().

2.1.2 Configuration Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::configuration::api::IBundleConfiguration`.

2.1.2.1 Ausführliche Beschreibung

Definiert in Zeile 10 der Datei Configuration.java.

Öffentliche Methoden

- `boolean bundleHasObjects ()`
- `boolean bundleHasRelations ()`
- `boolean bundleHasViews ()`
- `Document getObjectsJdomDocument ()` throws `JDOMException`, `IOException`

2.1.2.2 Dokumentation der Elementfunktionen

boolean bundleHasObjects ()

Definiert in Zeile 13 der Datei Configuration.java.

boolean bundleHasRelations ()

Definiert in Zeile 28 der Datei Configuration.java.

boolean bundleHasViews ()

Definiert in Zeile 18 der Datei Configuration.java.

Document `getObjectsJdomDocument ()` throws `JDOMException`, `IOException`

Definiert in Zeile 23 der Datei `Configuration.java`.

2.1.3 AbstractBundleManager Klassenreferenz**2.1.3.1 Ausführliche Beschreibung**

Definiert in Zeile 9 der Datei `AbstractBundleManager.java`.

Öffentliche Methoden

- `abstract Bundle getDetails ()`
- `abstract ArrayList< Bundle > getEamBundles ()`
- `javax.faces.component.html.HtmlInputHidden getNummer ()`
- `String getRemoveId ()`
- `abstract ArrayList< Bundle > getSystemBundles ()`
- `String getUrl ()`
- `abstract String installFromFile (String bundle)`
- `abstract String installFromUrl ()`
- `abstract String removeById ()`
- `void setNummer (javax.faces.component.html.HtmlInputHidden nummer)`
- `void setRemoveId (String removeId)`
- `void setUrl (String url)`
- `abstract String start ()`
- `abstract String stop ()`

Geschützte Attribute

- `javax.faces.component.html.HtmlInputHidden nummer`
- `String removeId`
- `String url`

2.1.3.2 Dokumentation der Elementfunktionen

`abstract Bundle getDetails ()` [pure virtual]

Autor:

Christian Zillmann Liefert die Details zu einem Bundle

Rückgabe:

ein Objekt vom Typ `Bundle`

abstract ArrayList<Bundle> getEamBundles () [pure virtual]

Autor:

Christian Zillmann Diese Methode liefert alle EAM-Bundles zurueck, also solche die das EAM-System erweitern.

Rückgabe:

ArrayList<Bundle>

javax.faces.component.html.HtmlInputHidden getNummer ()

Autor:

Christian Zillmann Liefert die Nummer des Hidden-Fields zurueck.

Rückgabe:

Definiert in Zeile 105 der Datei AbstractBundleManager.java.

Benutzt AbstractBundleManager.nummer.

String getRemoveld ()

Definiert in Zeile 32 der Datei AbstractBundleManager.java.

Benutzt AbstractBundleManager.removeId.

abstract ArrayList<Bundle> getSystemBundles () [pure virtual]

Autor:

Christian Zillmann Diese Methode liefert eine ArrayList mit allen Systembundles zurueck. Diese Bundles sollten im Normalfall nicht gestoppt werden.

Rückgabe:

ArrayList<Bunlde>

String getUrl ()

Definiert in Zeile 119 der Datei AbstractBundleManager.java.

Benutzt AbstractBundleManager.url.

abstract String installFromFile (String bundle) [pure virtual]

Autor:

Christian Zillmann Diese Methode installiert ein Bundle was hochgeladen wurden.

Parameter:

bundle

Rückgabe:

abstract String installFromUrl () [pure virtual]

Autor:

Christian Zillmann Diese Methode installiert ein Bundle von einer URL. return String mit Url

abstract String removeById () [pure virtual]

Autor:

Christian Zillmann Entfernt ein Bundle

Rückgabe:

void setNummer (javax.faces.component.html.HtmlInputHidden nummer)

Autor:

Christian Zillmann Setzt das Hidden-Field.

Rückgabe:

Definiert in Zeile 115 der Datei AbstractBundleManager.java.

Benutzt AbstractBundleManager.nummer.

void setRemoveld (String removeld)

Definiert in Zeile 36 der Datei AbstractBundleManager.java.

void setUrl (String url)

Definiert in Zeile 123 der Datei AbstractBundleManager.java.

abstract String start () [pure virtual]

Autor:

Christian Zillmann Diese Methode startet das Bundle mit der ID aus dem Hidden-Feld.

Rückgabe:

String wie der Workflow weitergeht.

Ausnahmebehandlung:

NumberFormatException

BundleException

abstract String stop () [pure virtual]

Autor:

Christian Zillmann Diese Methode stoppt das Bundle mit der ID aus dem Hidden-Feld.

Rückgabe:

String wie der Workflow weitergeht.

Ausnahmebehandlung:

NumberFormatException

BundleException

2.1.3.3 Dokumentation der Datenelemente

javax.faces.component.html.HtmlInputHidden nummer [protected]

Autor:

Christian Zillmann An dieses Feld wird das versteckte Feld gebunden, mit dem bestimmt wird, welches Bundle gestartet bzw. gestoppt wird.

Definiert in Zeile 15 der Datei AbstractBundleManager.java.

Wird benutzt von AbstractBundleManager.getNummer() und AbstractBundleManager.setNummer().

String removeId [protected]

Autor:

Christian Zillmann An diesen String wird die ID gebunden, vom dem Bundle, was entfernt werden soll.

Definiert in Zeile 30 der Datei AbstractBundleManager.java.

Wird benutzt von AbstractBundleManager.getRemoveId().

String url [protected]

Autor:

Christian Zillmann Dieser String enthaelt die URL von der ein Bundle installiert werden soll.

Definiert in Zeile 23 der Datei AbstractBundleManager.java.

Wird benutzt von AbstractBundleManager.getUrl().

2.1.4 Activator Klassenreferenz

Abgeleitet von org::osgi::framework::BundleActivator.

2.1.4.1 Ausführliche Beschreibung

Definiert in Zeile 39 der Datei Activator.java.

Öffentliche Methoden

- void start (BundleContext context) throws Exception
- void stop (BundleContext context) throws Exception

Statische öffentliche Attribute

- static final String bundleId = "bundleManager"
- static String bundleSymbolicName
- static String bundleVersion
- static long equinoxBundleId
- static final String PATH = "/bundleManager"

Private Attribute

- ServiceTracker httpServiceTracker

Statische private Attribute

- static final String WEB_ROOT = "/WebRoot"

Klassen

- class HttpServiceTracker
- class ServletContextListenerServletAdaptor

2.1.4.2 Dokumentation der Elementfunktionen

void start (BundleContext context) throws Exception

Definiert in Zeile 51 der Datei Activator.java.

Benutzt Activator.bundleSymbolicName, Activator.bundleVersion, Activator.equinoxBundleId und Activator.httpServiceTracker.

void stop (BundleContext context) throws Exception

Definiert in Zeile 67 der Datei Activator.java.

Benutzt Activator.httpServiceTracker.

2.1.4.3 Dokumentation der Datenelemente

final String bundleId = "bundleManager" [static]

Definiert in Zeile 43 der Datei Activator.java.

String bundleSymbolicName [static]

Definiert in Zeile 45 der Datei Activator.java.

Wird benutzt von Activator.start().

String bundleVersion [static]

Definiert in Zeile 46 der Datei Activator.java.

Wird benutzt von Activator.start().

long equinoxBundleId [static]

Definiert in Zeile 47 der Datei Activator.java.

Wird benutzt von Activator.start().

ServiceTracker httpServiceTracker [private]

Definiert in Zeile 49 der Datei Activator.java.

Wird benutzt von Activator.start() und Activator.stop().

final String PATH = "/bundleManager" [static]

Definiert in Zeile 41 der Datei Activator.java.

Wird benutzt von Activator.HttpServiceTracker.addingService() und Activator.HttpServiceTracker.removedService().

final String WEB_ROOT = "/WebRoot" [static, private]

Definiert in Zeile 42 der Datei Activator.java.

Wird benutzt von Activator.HttpServiceTracker.addingService().

2.1.5 Activator.HttpServiceTracker Klassenreferenz

Abgeleitet von org::osgi::util::tracker::ServiceTracker.

2.1.5.1 Ausführliche Beschreibung

Definiert in Zeile 74 der Datei Activator.java.

Öffentliche Methoden

- Object addingService (ServiceReference reference)
- HttpServiceTracker (BundleContext context)
- void removedService (ServiceReference reference, Object service)

2.1.5.2 Beschreibung der Konstruktoren und Destruktoren

HttpServiceTracker (BundleContext context)

Definiert in Zeile 77 der Datei Activator.java.

2.1.5.3 Dokumentation der Elementfunktionen

Object addingService (ServiceReference reference)

Definiert in Zeile 82 der Datei Activator.java.

Benutzt Activator.PATH und Activator.WEB_ROOT.

void removedService (ServiceReference reference, Object service)

Definiert in Zeile 139 der Datei Activator.java.

Benutzt Activator.PATH.

2.1.6 Activator.ServletContextListenerServletAdaptor Klassenreferenz

Abgeleitet von javax::servlet::Servlet.

2.1.6.1 Ausführliche Beschreibung

Definiert in Zeile 157 der Datei Activator.java.

Öffentliche Methoden

- void destroy ()
- ServletConfig getServletConfig ()
- String getServletInfo ()
- void init (ServletConfig config) throws ServletException
- void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException
- ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Private Attribute

- ServletConfig config
- Servlet delegate
- ClassLoader jspLoader
- ServletContextListener listener

2.1.6.2 Beschreibung der Konstruktoren und Destruktoren

ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Definiert in Zeile 169 der Datei Activator.java.

2.1.6.3 Dokumentation der Elementfunktionen

void destroy ()

Definiert in Zeile 237 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config, Activator.ServletContextListenerServletAdaptor.delegate, Activator.ServletContextListenerServletAdaptor.jspLoader und Activator.ServletContextListenerServletAdaptor.listener.

ServletConfig getServletConfig ()

Definiert in Zeile 265 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config.

String getServletInfo ()

Definiert in Zeile 273 der Datei Activator.java.

void init (ServletConfig config) throws ServletException

Definiert in Zeile 185 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config,
Activator.ServletContextListenerServletAdaptor.delegate,
Activator.ServletContextListenerServletAdaptor.jspLoader und
Activator.ServletContextListenerServletAdaptor.listener.

void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException

Definiert in Zeile 213 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.delegate und
Activator.ServletContextListenerServletAdaptor.jspLoader.

2.1.6.4 Dokumentation der Datenelemente**ServletConfig config [private]**

Definiert in Zeile 159 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.getServletConfig() und
Activator.ServletContextListenerServletAdaptor.init().

Servlet delegate [private]

Definiert in Zeile 163 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.init() und
Activator.ServletContextListenerServletAdaptor.service().

ClassLoader jspLoader [private]

Definiert in Zeile 165 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.init() und
Activator.ServletContextListenerServletAdaptor.service().

ServletContextListener listener [private]

Definiert in Zeile 161 der Datei Activator.java.

Wird benutzt von `Activator.ServletContextListenerServletAdaptor.destroy()` und `Activator.ServletContextListenerServletAdaptor.init()`.

2.1.7 BundleManager Klassenreferenz

2.1.7.1 Ausführliche Beschreibung

Autor:

zillmann TODO Neue Bundles die installiert werden, muessen beim naechsten Start des Servers noch mitgestartet werden. TODO Einige Fehler muessen noch mit entsprechenden Exceptions abgefangen werden.

Definiert in Zeile 52 der Datei `BundleManager.java`.

Öffentliche Methoden

- `BundleManager ()`
- `void checkDeleteBundle ()`
- `String configureBundle ()` throws `DataConversionException`, `NoElementsFoundException`, `ValueNotFoundException`, `ParseException`
- `void deleteWithDependencies ()`
- `void deleteWithoutDependencies ()`
- `LinkedList< Bundle > getBundleDependencies ()`
- `HtmlInputHidden getBundleName ()`
- `String getBundleSymbolicName ()`
- `ArrayList< EAMBundle > getClearedEAMBundles ()`
- `Integer getDeleteBundleId ()`
- `Bundle getDetails ()`
- `ArrayList< EAMBundle > getEamBundles ()`
- `HtmlInputHidden getManagerModuleName ()`
- `javax.faces.component.html.HtmlInputHidden getNummer ()`
- `String getRemoveId ()`
- `ArrayList< Bundle > getSystemBundles ()`
- `UIInput getTestName ()`
- `String getUrl ()`
- `String installFromUrl ()`
- `boolean isBundleConfigurable ()`
- `boolean isBundleDependenciesEmpty ()`
- `boolean isDeleteBundle ()`
- `void noDelete ()`
- `String removeById ()`

- void setBundleDependencies (LinkedList< Bundle > bundleDependencies)
- void setBundleDependenciesEmpty (boolean bundleDependenciesEmpty)
- void setBundleName (HtmlInputHidden bundleName)
- void setBundleSymbolicName (String bundleSymbolicName)
- void setDeleteBundle (boolean deleteBundle)
- void setDeleteBundleId (Integer deleteBundleId)
- void setManagerModuleName (HtmlInputHidden managerModuleName)
- void setNummer (javax.faces.component.html.HtmlInputHidden nummer)
- void setRemoveId (String removeId)
- void setTestName (UIInput testName)
- void setUrl (String url)
- String show () throws NumberFormatException, BundleException
- String shutDown () throws NumberFormatException, BundleException
- String start () throws NumberFormatException, BundleException
- String stop () throws NumberFormatException, BundleException

Öffentliche, statische Methoden

- static String installFromFile (String bundle)

Öffentliche Attribute

- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Private Methoden

- void handleException (Exception ex)
- void removeBundleDependingEntries (int equinoxId)
- void removeBundleDependingEntries (String bundleName)

Private Attribute

- LinkedList< Bundle > bundleDependencies
- boolean bundleDependenciesEmpty = false
- HtmlInputHidden bundleName
- String bundleSymbolicName
- HashMap< String, EAMConfigGeneratorJDOM > configGeneratorStore

- boolean deleteBundle = false
- Integer deleteBundleId
- DAOFactoryStaticClass factory = DAOFactoryStaticClass.getInstance(userId, password, this.getClass().getName())
- HtmlInputHidden managerModuleName
- HtmlInputHidden nummer
- String removeId
- UIInput testName
- String url

2.1.7.2 Beschreibung der Konstruktoren und Destruktoren

BundleManager ()

Definiert in Zeile 97 der Datei BundleManager.java.

2.1.7.3 Dokumentation der Elementfunktionen

void checkDeleteBundle ()

Definiert in Zeile 252 der Datei BundleManager.java.

Benutzt BundleManager.bundleDependencies, BundleManager.bundleDependenciesEmpty, BundleManager.bundleSymbolicName, BundleManager.deleteBundle, BundleManager.deleteBundleId, BundleManager.getEamBundles() und BundleManager.password.

String configureBundle () throws DataConversionException, NoElementsFoundException, ValueNotFoundException, ParseException

Ermittelt nach Betätigen des Konfigurieren-Buttons in der Bundle-Overview den weiteren Verlauf des Konfigurationsprozesses. dazu wird geprüft, ob ein Bundle EAM-Objekte, EAM-Relationen oder Rollen hat. Abhängig verändert sich das Menu während des Config-Prozesses und die Startseite. Sind Rollen vorhanden, wird immer auf der Seite des Rollenmappings begonnen, sonst auf der Objekt-Mapping-Seite.

Rückgabe:

Der String einer JSF-Seite, auf der der Konfigurations-Vorgang fortgesetzt werden soll.

Ausnahmebehandlung:

Exception Die Exceptions werden durch das Auswerten der verschiedenen XML-Dateien geworfen und bis zu dieser Methode weitergeleitet. Dadurch kann an dieser zentralen Stelle auf diese reagiert werden und abhängig von der Art der Exception eine entsprechende Mitteilung auf der Bundle-Overview-Seite ausgegeben werden.

Definiert in Zeile 617 der Datei BundleManager.java.

Benutzt BundleManager.nummer, BundleManager.password und BundleManager.testName.

void deleteWithDependencies ()

Entfernt ein ausgewähltes Bundle inklusive der zu anderen Bundles bestehenden Abhängigkeiten. Dazu gehören z.B. verknüpfte oder gespeichert Objekte oder Relationen, die von einem anderen Bundle für seinen Betrieb verwendet werden.

Definiert in Zeile 330 der Datei BundleManager.java.

Benutzt BundleManager.bundleDependenciesEmpty, BundleManager.bundleSymbolicName, BundleManager.deleteBundle, BundleManager.deleteBundleId, BundleManager.factory, DAOFactoryStaticClass.objectDAO, BundleManager.password, DAOFactoryStaticClass.relationDAO und BundleManager.removeBundleDependingEntries().

void deleteWithoutDependencies ()

Entfernt ein Bundle ohne die Abhängigkeiten, die zu anderen Bundles bestehen.

Definiert in Zeile 427 der Datei BundleManager.java.

Benutzt BundleManager.bundleDependenciesEmpty, BundleManager.bundleSymbolicName, BundleManager.deleteBundle, BundleManager.deleteBundleId und BundleManager.removeBundleDependingEntries().

LinkedList<Bundle> getBundleDependencies ()

Definiert in Zeile 933 der Datei BundleManager.java.

Benutzt BundleManager.bundleDependencies.

HtmlInputHidden getBundleName ()

Definiert in Zeile 914 der Datei BundleManager.java.

Benutzt BundleManager.bundleName.

String getBundleSymbolicName ()

Definiert in Zeile 101 der Datei BundleManager.java.

Benutzt BundleManager.bundleSymbolicName.

ArrayList<EAMBundle> getClearedEAMBundles ()

Ermittelt eine Liste von Bundles, in denen der Core und der Bundlemanager nicht enthalten sind. Dies wird für die Bundleübersicht beim deinstallieren eines Bundles benötigt. Dadurch kann der Kern und der Bundlemanager nicht über die Webseite deinstalliert.

Rückgabe:

Die Liste aller Bundles ohne Kern und Bundlemanager.

Definiert in Zeile 895 der Datei BundleManager.java.

Benutzt BundleManager.getEamBundles().

Integer getDeleteBundleId ()

Definiert in Zeile 924 der Datei BundleManager.java.

Benutzt BundleManager.deleteBundleId.

Bundle getDetails ()

Liefert detailliertere Informationen zu einem ausgewählten Bundle. Setzt diese Informationen als Attribute im Request-Objekt des Faces-Context.

Rückgabe:

Das ausgewählte Bundle, dessen Informationen angezeigt werden sollen.

Definiert in Zeile 562 der Datei BundleManager.java.

Benutzt ParamsGenerator.getDependedBundle().

ArrayList<EAMBundle> getEamBundles ()

Autor:

Christian Zillmann Diese Methode liefert alle EAM-Bundles zurueck, also solche die das EAM-System erweitern.

Rückgabe:

ArrayList<Bundle>

Definiert in Zeile 156 der Datei BundleManager.java.

Wird benutzt von BundleManager.checkDeleteBundle() und BundleManager.getClearedEAMBundles().

HtmlInputHidden getManagerModuleName ()

Definiert in Zeile 949 der Datei BundleManager.java.

Benutzt BundleManager.managerModuleName.

javax.faces.component.html.HtmlInputHidden getNummer ()

Liefert die Nummer des Hidden-Fields zurueck.

Rückgabe:

Definiert in Zeile 536 der Datei BundleManager.java.

Benutzt BundleManager.nummer.

String getRemoveld ()

Definiert in Zeile 121 der Datei BundleManager.java.

Benutzt BundleManager.removeId.

ArrayList<Bundle> getSystemBundles ()**Autor:**

Christian Zillmann Diese Methode liefert eine ArrayList mit allen Systembundles zurueck. Diese Bundles sollten im Normalfall nicht gestoppt werden.

Rückgabe:

ArrayList<Bunlde>

Definiert in Zeile 136 der Datei BundleManager.java.

UIInput getTestName ()

Definiert in Zeile 957 der Datei BundleManager.java.

Benutzt BundleManager.testName.

String getUrl ()

Definiert in Zeile 549 der Datei BundleManager.java.

Benutzt BundleManager.url.

void handleException (Exception ex) [private]

Dient dem Handling und Verarbeiten der Exceptions, die in configureBundle geworfen werden.

Parameter:

ex Die Exception, die verarbeitet werden soll.

Definiert in Zeile 816 der Datei BundleManager.java.

static String installFromFile (String bundle) [static]

Autor:

Christian Zillmann Installiert ein Bundle von einem File

Parameter:

bundle

Rückgabe:

Definiert in Zeile 236 der Datei BundleManager.java.

String installFromUrl ()

Autor:

Christian Zillmann Installiert ein Bundle von einer URL

Rückgabe:

Definiert in Zeile 216 der Datei BundleManager.java.

Benutzt BundleManager.url.

boolean isBundleConfigurable ()

Überprüft, ob ein Bundle konfigurierbar ist. Dadurch wird in der View der Button "Konfigurieren" dargestellt, wodurch die Konfiguration eines Bundles durchgeführt werden kann.

Rückgabe:

True, wenn ein Bundle Konfigurierbar ist, sonst false.

Definiert in Zeile 780 der Datei BundleManager.java.

Benutzt BundleManager.nummer.

boolean isBundleDependenciesEmpty ()

Definiert in Zeile 941 der Datei BundleManager.java.

Benutzt BundleManager.bundleDependenciesEmpty.

boolean isDeleteBundle ()

Definiert in Zeile 111 der Datei BundleManager.java.

Benutzt BundleManager.deleteBundle.

void noDelete ()

Setzt alle Boolean-Flags, die beim Entfernen eines Bundles benötigt werden, wieder auf ihren Standard-Wert zurück.

Definiert in Zeile 316 der Datei BundleManager.java.

Benutzt BundleManager.bundleDependencies, BundleManager.bundleDependenciesEmpty, BundleManager.bundleSymbolicName, BundleManager.deleteBundle und BundleManager.deleteBundleId.

void removeBundleDependingEntries (int equinoxId) [private]

Übernimmt das Entfernen der Views, der Modul-Methoden und der zum Bundle gehörenden Verknüpfungseinträge in den jeweiligen Tabellen.

Parameter:

equinoxId Die Equinox-Id des Bundles, zu dem die jeweiligen Einträge entfernt werden sollen.

Definiert in Zeile 483 der Datei BundleManager.java.

Benutzt BundleManager.password.

void removeBundleDependingEntries (String bundleName) [private]

Übernimmt das Entfernen der Views, der Modul-Methoden und der zum Bundle gehörenden Verknüpfungseinträge in den jeweiligen Tabellen.

Parameter:

bundleName Der Name des Bundles, zu dem die jeweiligen Einträge entfernt werden sollen.

Definiert in Zeile 454 der Datei BundleManager.java.

Benutzt BundleManager.password.

Wird benutzt von BundleManager.deleteWithDependencies() und BundleManager.deleteWithoutDependencies().

String removeById ()

Entfernt mit Hilfe einer in der View gesetzten Bundle-Id das gewünschte Bundle.

Rückgabe:

Definiert in Zeile 513 der Datei BundleManager.java.

Benutzt BundleManager.removeId.

void setBundleDependencies (LinkedList< Bundle > bundleDependencies)

Definiert in Zeile 937 der Datei BundleManager.java.

void setBundleDependenciesEmpty (boolean bundleDependenciesEmpty)

Definiert in Zeile 945 der Datei BundleManager.java.

void setBundleName (HtmlInputHidden bundleName)

Definiert in Zeile 919 der Datei BundleManager.java.

void setBundleSymbolicName (String bundleSymbolicName)

Definiert in Zeile 106 der Datei BundleManager.java.

void setDeleteBundle (boolean deleteBundle)

Definiert in Zeile 116 der Datei BundleManager.java.

void setDeleteBundleId (Integer deleteBundleId)

Definiert in Zeile 929 der Datei BundleManager.java.

void setManagerModuleName (HtmlInputHidden managerModuleName)

Definiert in Zeile 953 der Datei BundleManager.java.

void setNummer (javax.faces.component.html.HtmlInputHidden nummer)

Setzt das Hidden-Field.

Rückgabe:

Definiert in Zeile 545 der Datei BundleManager.java.

Benutzt BundleManager.nummer.

void setRemoveld (String removeld)

Definiert in Zeile 125 der Datei BundleManager.java.

void setTestName (UIInput testName)

Definiert in Zeile 961 der Datei BundleManager.java.

void setUrl (String url)

Definiert in Zeile 553 der Datei BundleManager.java.

String show () throws NumberFormatException, BundleException

Definiert in Zeile 208 der Datei BundleManager.java.

String shutDown () throws NumberFormatException, BundleException

Stoppt das Bundle, das für die Funktionsfähigkeit des Frameworks verantwortlich ist.

Rückgabe:

Die Seite, zu der nach dem Abschalten weitergeleitet werden soll, als String für faces-config.

Ausnahmebehandlung:

NumberFormatException

BundleException

Definiert in Zeile 594 der Datei BundleManager.java.

String start () throws NumberFormatException, BundleException**Autor:**

Christian Zillmann Diese Methode startet das Bundle mit der ID aus dem Hidden-Feld.

Rückgabe:

String wie der Workflow weitergeht.

Ausnahmebehandlung:

NumberFormatException

BundleException

Definiert in Zeile 198 der Datei BundleManager.java.

Benutzt BundleManager.nummer.

String stop () throws NumberFormatException, BundleException**Autor:**

Christian Zillmann Diese Methode stoppt das Bundle mit der ID aus dem Hidden-Feld.

Rückgabe:

String wie der Workflow weitergeht.

Ausnahmebehandlung:

NumberFormatException

BundleException

Definiert in Zeile 179 der Datei BundleManager.java.

Benutzt BundleManager.nummer.

2.1.7.4 Dokumentation der Datenelemente

LinkedList<Bundle> bundleDependencies [private]

Definiert in Zeile 92 der Datei BundleManager.java.

Wird benutzt von BundleManager.checkDeleteBundle(),
BundleManager.getBundleDependencies() und BundleManager.noDelete().

boolean bundleDependenciesEmpty = false [private]

Definiert in Zeile 94 der Datei BundleManager.java.

Wird benutzt von BundleManager.checkDeleteBundle(),
BundleManager.deleteWithDependencies(), BundleManager.deleteWithoutDependencies(),
BundleManager.isBundleDependenciesEmpty() und BundleManager.noDelete().

HtmlInputHidden bundleName [private]

Definiert in Zeile 66 der Datei BundleManager.java.

Wird benutzt von BundleManager.getBundleName().

String bundleSymbolicName [private]

Definiert in Zeile 89 der Datei BundleManager.java.

Wird benutzt von BundleManager.checkDeleteBundle(),
BundleManager.deleteWithDependencies(), BundleManager.deleteWithoutDependencies(),
BundleManager.getBundleSymbolicName() und BundleManager.noDelete().

HashMap<String, EAMConfigGeneratorJDOM> configGeneratorStore [private]

Definiert in Zeile 84 der Datei BundleManager.java.

boolean deleteBundle = false [private]

Definiert in Zeile 86 der Datei BundleManager.java.

Wird benutzt von BundleManager.checkDeleteBundle(), BundleManager.deleteWithDependencies(), BundleManager.deleteWithoutDependencies(), BundleManager.isDeleteBundle() und BundleManager.noDelete().

Integer deleteBundleId [private]

Definiert in Zeile 87 der Datei BundleManager.java.

Wird benutzt von BundleManager.checkDeleteBundle(), BundleManager.deleteWithDependencies(), BundleManager.deleteWithoutDependencies(), BundleManager.getDeleteBundleId() und BundleManager.noDelete().

DAOFactoryStaticClass factory = DAOFactoryStaticClass.getInstance(userId, password, this.getClass().getName()) [private]

Definiert in Zeile 58 der Datei BundleManager.java.

Wird benutzt von BundleManager.deleteWithDependencies().

HtmlInputHidden managerModuleName [private]

Definiert in Zeile 68 der Datei BundleManager.java.

Wird benutzt von BundleManager.getManagerModuleName().

HtmlInputHidden nummer [private]

Definiert in Zeile 64 der Datei BundleManager.java.

Wird benutzt von BundleManager.configureBundle(), BundleManager.getNummer(), BundleManager.isBundleConfigurable(), BundleManager.setNummer(), BundleManager.start() und BundleManager.stop().

String password = (String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")

Definiert in Zeile 56 der Datei BundleManager.java.

Wird benutzt von BundleManager.checkDeleteBundle(), BundleManager.configureBundle(), BundleManager.deleteWithDependencies() und BundleManager.removeBundleDependingEntries().

String removeId [private]

Definiert in Zeile 82 der Datei BundleManager.java.

Wird benutzt von BundleManager.getRemoveId() und BundleManager.removeById().

UInput testName [private]

Definiert in Zeile 70 der Datei BundleManager.java.

Wird benutzt von BundleManager.configureBundle() und BundleManager.getTestName().

String url [private]

Definiert in Zeile 76 der Datei BundleManager.java.

Wird benutzt von BundleManager.getUrl() und BundleManager.installFromUrl().

String userId =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Definiert in Zeile 55 der Datei BundleManager.java.

2.1.8 Context Klassenreferenz

2.1.8.1 Ausführliche Beschreibung

Autor:

Christian Zillmann Speichert den BundleContext des Bundlemanagers. Wird von der Activator Klasse gesetzt.

Definiert in Zeile 11 der Datei Context.java.

Öffentliche, statische Methoden

- static BundleContext getContext ()
- static void setContext (BundleContext context)

Statische private Attribute

- static BundleContext context

2.1.8.2 Dokumentation der Elementfunktionen

static BundleContext getContext () [static]

Definiert in Zeile 16 der Datei Context.java.

Benutzt Context.context.

static void setContext (BundleContext context) [static]

Definiert in Zeile 21 der Datei Context.java.

2.1.8.3 Dokumentation der Datenelemente

BundleContext context [static, private]

Definiert in Zeile 14 der Datei Context.java.

Wird benutzt von Context.getContext().

2.2 Dependency

2.2.1 Bundle Klassenreferenz

2.2.1.1 Ausführliche Beschreibung

Die Klasse Bundle repräsentiert ein Bundle.

Autor:

Yu Liu

Siehe auch:

DependencyApplet

Definiert in Zeile 12 der Datei Bundle.java.

Öffentliche Methoden

- boolean addDependedBundle (Bundle bundle)
- Bundle (String symbolicName)
- Set< Bundle > getDependedBundles ()
- String getName ()
- boolean removeDependedBundle (Bundle bundle)
- void setDependedBundles (Set< Bundle > dependedBundles)
- void setName (String name)

Private Attribute

- Set< Bundle > dependedBundles = new HashSet<Bundle>()
- String name

2.2.1.2 Beschreibung der Konstruktoren und Destruktoren

Bundle (String symbolicName)

Konstruktor

Parameter:

symbolicName Name des Bundle

Definiert in Zeile 30 der Datei Bundle.java.

Benutzt Bundle.name.

2.2.1.3 Dokumentation der Elementfunktionen

boolean addDependedBundle (Bundle bundle)

Fügt ein abgehangenes Bundle hinzu.

Parameter:

bundle ein abgehangenes Bundle

Rückgabe:

true, falls das Bundle erfolgreich hinzugefügt ist.

Definiert in Zeile 79 der Datei Bundle.java.

Benutzt Bundle.dependedBundles.

Set<Bundle> getDependedBundles ()

Liefert die abgehangenen Bundles zurück.

Rückgabe:

eine Menge von Bundles

Definiert in Zeile 39 der Datei Bundle.java.

Benutzt Bundle.dependedBundles.

Wird benutzt von DependencyPanel.initializeGraph().

String getName ()

Liefert den Namen des Bundle zurück.

Rückgabe:

der Name eines Bundle

Definiert in Zeile 58 der Datei Bundle.java.

Benutzt Bundle.name.

Wird benutzt von DependencyPanel.setRoot().

boolean removeDependedBundle (Bundle bundle)

Entfernt ein abgehangenes Bundle.

Parameter:

bundle ein abgehangenes Bundle

Rückgabe:

true, falls das Bundle erfolgreich entfernt ist.

Definiert in Zeile 90 der Datei Bundle.java.

Benutzt Bundle.dependedBundles.

void setDependedBundles (Set< Bundle > dependedBundles)

Setter der abgehangenen Bundles

Parameter:

dependedBundles eine Menge von Bundles

Definiert in Zeile 49 der Datei Bundle.java.

void setName (String name)

Setter des Bundle's Namens

Parameter:

name der Name eines Bundle

Definiert in Zeile 68 der Datei Bundle.java.

2.2.1.4 Dokumentation der Datenelemente

Set<Bundle> dependedBundles = new HashSet<Bundle>() [private]

die von diesem Bundle abgehangenen Bundles

Definiert in Zeile 22 der Datei Bundle.java.

Wird benutzt von Bundle.addDependedBundle(), Bundle.getDependedBundles() und Bundle.removeDependedBundle().

String name [private]

Name eines Bundle

Definiert in Zeile 17 der Datei Bundle.java.

Wird benutzt von Bundle.Bundle() und Bundle.getName().

2.2.2 BundleConnection Klassenreferenz

2.2.2.1 Ausführliche Beschreibung

Die Klasse BundleConnection repräsentiert die Abhängigkeit zwischen zwei Bundles.

Autor:

Yu Liu

Definiert in Zeile 12 der Datei BundleConnection.java.

Öffentliche Methoden

- BundleConnection (BundleNode begin, BundleNode end)
- BundleNode getBegin ()
- Color getColor ()
- BundleNode getEnd ()
- void setBegin (BundleNode begin)
- void setColor (Color color)
- void setEnd (BundleNode end)

Private Attribute

- BundleNode begin = null
- Color color = null
- BundleNode end = null

2.2.2.2 Beschreibung der Konstruktoren und Destruktoren

BundleConnection (BundleNode begin, BundleNode end)

Konstruktor

Parameter:

begin Quelle

end Ziel

Definiert in Zeile 34 der Datei BundleConnection.java.

Benutzt BundleConnection.color.

2.2.2.3 Dokumentation der Elementfunktionen

BundleNode getBegin ()

Liefert die Quelle einer Connection zurück.

Rückgabe:

ein `BundleNode`

Definiert in Zeile 44 der Datei `BundleConnection.java`.

Benutzt `BundleConnection.begin`.

Wird benutzt von `BundleNode.BundleNodeMouseMotion.mouseDragged()` und `BundleNode.BundleNodeMouseProcess.mouseReleased()`.

Color getColor ()

Liefert die Farbe einer Connection zurück.

Rückgabe:

die Farbe

Definiert in Zeile 52 der Datei `BundleConnection.java`.

Benutzt `BundleConnection.color`.

BundleNode getEnd ()

Liefert das Ziel einer Connection zurück.

Rückgabe:

Definiert in Zeile 60 der Datei `BundleConnection.java`.

Benutzt `BundleConnection.end`.

Wird benutzt von `BundleNode.BundleNodeMouseMotion.mouseDragged()` und `BundleNode.BundleNodeMouseProcess.mouseReleased()`.

void setBegin (BundleNode begin)

Setter der Quelle einer Connection

Parameter:

begin ein `BundleNode`

Definiert in Zeile 68 der Datei `BundleConnection.java`.

void setColor (Color color)

Setter der Farbe einer Connection

Parameter:

color eine neue Farbe

Definiert in Zeile 76 der Datei BundleConnection.java.

Wird benutzt von BundleNode.BundleNodeMouseMotion.mouseDragged() und BundleNode.BundleNodeMouseProcess.mouseReleased().

void setEnd (BundleNode end)

Setter des Ziels einer Connection

Parameter:

end ein BundleNode

Definiert in Zeile 84 der Datei BundleConnection.java.

2.2.2.4 Dokumentation der Datenelemente

BundleNode begin = null [private]

die Quelle einer Connection

Definiert in Zeile 17 der Datei BundleConnection.java.

Wird benutzt von BundleConnection.getBegin().

Color color = null [private]

die Farbe einer Connection

Definiert in Zeile 22 der Datei BundleConnection.java.

Wird benutzt von BundleConnection.BundleConnection() und BundleConnection.getColor().

BundleNode end = null [private]

das Ziel einer Connection

Definiert in Zeile 27 der Datei BundleConnection.java.

Wird benutzt von BundleConnection.getEnd().

2.2.3 BundleNode Klassenreferenz

Abgeleitet von javax.swing.JButton.

2.2.3.1 Ausführliche Beschreibung

Die Klasse `BundleNode` ist die Graphische Darstellung eines Bundle. Sie vererbt die Klasse `JButton`, damit sie verschiedene Ereignisse empfangen und behandeln kann.

Autor:

Yu Liu

Definiert in Zeile 26 der Datei `BundleNode.java`.

Öffentliche Methoden

- `boolean addConnection (BundleNode node)`
- `BundleNode (String symbolicName)`
- `Point getAnchorage (int part)`
- `Bundle getBundle ()`
- `Point getCenter ()`
- `Point getPartPoint (int part)`
- `int getQuarterWidth ()`
- `Set< BundleNode > getTargets ()`
- `void paint (Graphics g)`
- `void setBundle (Bundle bundle)`
- `void setTargets (Set< BundleNode > targets)`

Statische öffentliche Attribute

- `static final int BOTTOM_SIDE = 2`
- `static Set< BundleConnection > connections = new HashSet<BundleConnection>()`
- `static final int LEFT_SIDE = 3`
- `static final int RIGHT_SIDE = 1`
- `static final int TOP_SIDE = 0`

Private Attribute

- `Bundle bundle`
- `int focusx = 0`
- `int focusy = 0`
- `int quarterWidth`
- `Set< BundleNode > targets = new HashSet<BundleNode>()`

Statische private Attribute

- static final long serialVersionUID = 1L

Klassen

- class BundleNodeMouseMotion
- class BundleNodeMouseProcess

2.2.3.2 Beschreibung der Konstruktoren und Destruktoren

BundleNode (String symbolicName)

Konstruktor

Parameter:

symbolicName der Name eines BundleNode

Definiert in Zeile 189 der Datei BundleNode.java.

2.2.3.3 Dokumentation der Elementfunktionen

boolean addConnection (BundleNode node)

Hinzufügen eines abgehangenes BundleNode

Parameter:

node

Rückgabe:

true, falls das Node erfolgreich hinzugefügt ist. false, falls das Hinzufügen fehlerschlagt.

Definiert in Zeile 206 der Datei BundleNode.java.

Benutzt BundleNode.targets.

Wird benutzt von DependencyPanel.initializeGraph().

Point getAnchorage (int part)

Definiert in Zeile 269 der Datei BundleNode.java.

Benutzt BundleNode.BOTTOM_SIDE, BundleNode.getQuarterWidth(), BundleNode.LEFT_SIDE, BundleNode.quarterWidth, BundleNode.RIGHT_SIDE und BundleNode.TOP_SIDE.

Bundle getBundle ()

Liefert Bundle zurück.

Rückgabe:

das vom Bundelnode repräsentierte Bundle

Definiert in Zeile 222 der Datei BundleNode.java.

Benutzt BundleNode.bundle.

Point getCenter ()

Liefert das geometrisches Zentralpunkt eines BundleNode zurück.

Rückgabe:

ein Point

Definiert in Zeile 231 der Datei BundleNode.java.

Wird benutzt von DependencyPanel.paint().

Point getPartPoint (int part)

Liefert das Mittelpunkt einer Seite zurück.

Parameter:

part Seite

Rückgabe:

ein Point

Definiert in Zeile 243 der Datei BundleNode.java.

Benutzt BundleNode.BOTTOM_SIDE, BundleNode.LEFT_SIDE, BundleNode.RIGHT_SIDE und BundleNode.TOP_SIDE.

Wird benutzt von GraphicsUtils.getContainedPoints().

int getQuarterWidth ()

Liefert das Viertel der Breite zurück.

Rückgabe:

Definiert in Zeile 265 der Datei BundleNode.java.

Benutzt BundleNode.quarterWidth.

Wird benutzt von BundleNode.getAnchorage().

Set<BundleNode> getTargets ()

Liefert Zielknoten zurück.

Rückgabe:

eine Menge von Zielknoten

Definiert in Zeile 293 der Datei BundleNode.java.

Benutzt BundleNode.targets.

void paint (Graphics g)

Malen des Bundlenode

Definiert in Zeile 301 der Datei BundleNode.java.

Wird benutzt von DependencyPanel.paint().

void setBundle (Bundle bundle)

Setter des Bundle

Parameter:

bundle ein neues Bundle

Definiert in Zeile 309 der Datei BundleNode.java.

Wird benutzt von DependencyPanel.initializeGraph() und DependencyPanel.setRoot().

void setTargets (Set< BundleNode > targets)

Setter der Zielknoten

Parameter:

targets die neuen Zielknoten

Definiert in Zeile 317 der Datei BundleNode.java.

2.2.3.4 Dokumentation der Datenelemente

final int BOTTOM_SIDE = 2 [static]

Untenseite

Definiert in Zeile 144 der Datei BundleNode.java.

Wird benutzt von BundleNode.getAnchorage() und BundleNode.getPartPoint().

Bundle bundle [private]

zu dargestelltes Bundle

Definiert in Zeile 166 der Datei BundleNode.java.

Wird benutzt von BundleNode.getBundle().

Set<BundleConnection> connections = new HashSet<BundleConnection>()
[static]

Connections eines BundleNode

Definiert in Zeile 139 der Datei BundleNode.java.

int focusx = 0 [private]

Definiert in Zeile 168 der Datei BundleNode.java.

int focusy = 0 [private]

Definiert in Zeile 168 der Datei BundleNode.java.

final int LEFT_SIDE = 3 [static]

Linkseite

Definiert in Zeile 149 der Datei BundleNode.java.

Wird benutzt von BundleNode.getAnchorage() und BundleNode.getPartPoint().

int quarterWidth [private]

das Viertel der Breite

Definiert in Zeile 173 der Datei BundleNode.java.

Wird benutzt von BundleNode.getAnchorage() und BundleNode.getQuarterWidth().

final int RIGHT_SIDE = 1 [static]

Rechtseite

Definiert in Zeile 154 der Datei BundleNode.java.

Wird benutzt von BundleNode.getAnchorage() und BundleNode.getPartPoint().

final long serialVersionUID = 1L [static, private]

Definiert in Zeile 161 der Datei BundleNode.java.

Set<BundleNode> targets = new HashSet<BundleNode>() [private]

abgehangene Bundles

Definiert in Zeile 178 der Datei BundleNode.java.

Wird benutzt von BundleNode.addConnection() und BundleNode.getTargets().

final int TOP_SIDE = 0 [static]

Obenseite

Definiert in Zeile 159 der Datei BundleNode.java.

Wird benutzt von BundleNode.getAnchorage() und BundleNode.getPartPoint().

2.2.4 BundleNode.BundleNodeMouseMotion Klassenreferenz

Abgeleitet von java::awt::event::MouseMotionAdapter.

2.2.4.1 Ausführliche Beschreibung

Die Klasse BundleNodeMouseMotion vererbt die Klasse MouseMotionAdapter, damit die Klasse BundleNode die Bewegung des Maus behandeln kann.

Autor:

Yu Liu

Definiert in Zeile 36 der Datei BundleNode.java.

Öffentliche Methoden

- void mouseDragged (MouseEvent e)

2.2.4.2 Dokumentation der Elementfunktionen

void mouseDragged (MouseEvent e)

Behandelt Schleppen des Maus.

Definiert in Zeile 42 der Datei BundleNode.java.

Benutzt BundleConnection.getBegin(), BundleConnection.getEnd() und BundleConnection.setColor().

2.2.5 BundleNode.BundleNodeMouseProcess Klassenreferenz

Abgeleitet von java::awt::event::MouseAdapter.

2.2.5.1 Ausführliche Beschreibung

Die Klasse BundleNodeMouseProcess vererbt die Klasse MouseAdapter, damit die Klasse BundleNode die Ereignisse wie Klicken behandeln kann.

Autor:

Yu Liu

Definiert in Zeile 85 der Datei BundleNode.java.

Öffentliche Methoden

- void mouseClicked (MouseEvent evt)
- void mousePressed (MouseEvent evt)
- void mouseReleased (MouseEvent e)

2.2.5.2 Dokumentation der Elementfunktionen**void mouseClicked (MouseEvent evt)**

behandelt das Klicken des Maus.

Definiert in Zeile 91 der Datei BundleNode.java.

Benutzt DependencyPanel.getRoot() und DependencyPanel.setRoot().

void mousePressed (MouseEvent evt)

Behandelt das Drücken des Maus.

Definiert in Zeile 110 der Datei BundleNode.java.

void mouseReleased (MouseEvent e)

Behandelt das Befreien des Maus.

Definiert in Zeile 123 der Datei BundleNode.java.

Benutzt BundleConnection.getBegin(), BundleConnection.getEnd() und BundleConnection.setColor().

2.2.6 DependencyApplet Klassenreferenz

Abgeleitet von javax::swing::JApplet.

2.2.6.1 Ausführliche Beschreibung

Die Klasse `DependencyApplet` vererbt die Klasse `JApplet`, damit die Bundleabhängigkeiten in einer Webseite dargestellt werden können. Die Klasse liest zuerst alle Parameter ein und analysiert sie, damit es ausgerechnet wird, auf welcher Ebene befindet sich ein Node und mit welchen anderen Bundles ein Node verbindet.

Autor:

Yu Liu

Definiert in Zeile 26 der Datei DependencyApplet.java.

Öffentliche Methoden

- Bundle getRoot ()
- void init ()
- void setDefaultSize ()

Private Attribute

- Map< String, Set< String > > bundleLinks = new HashMap<String, Set<String>>()
- int counter = 0
- Map< String, Bundle > idBundle = new HashMap<String, Bundle>()
- Bundle root

Statische private Attribute

- static final Dimension DEFAULT_SIZE = new Dimension(1024, 480)
- static final long serialVersionUID = 1L

2.2.6.2 Dokumentation der Elementfunktionen

Bundle getRoot ()

Liefert den Wurzelt des Baums zurück.

Rückgabe:

Definiert in Zeile 60 der Datei DependencyApplet.java.

Benutzt DependencyApplet.root.

void init ()

Initialisierung des Applets. Hier werden alle Parameter analysiert und ausgerechnet, auf welche Ebene befindet sich ein BundleNode und mit welchen anderen Nodes soll ein Node verbunden werden.

Definiert in Zeile 77 der Datei DependencyApplet.java.

Benutzt `DependencyApplet.bundleLinks`, `DependencyApplet.counter`, `DependencyApplet.idBundle`, `DependencyApplet.root` und `DependencyApplet.setDefaultSize()`.

void setDefaultSize ()

Stellt die defaulte Größe ein.

Definiert in Zeile 67 der Datei `DependencyApplet.java`.

Wird benutzt von `DependencyApplet.init()` und `DependencyPanel.setRoot()`.

2.2.6.3 Dokumentation der Datenelemente

Map<String, Set<String> > bundleLinks = new HashMap<String, Set<String>>() [private]

Links zwischen Knoten(Bundles)

Definiert in Zeile 38 der Datei `DependencyApplet.java`.

Wird benutzt von `DependencyApplet.init()`.

int counter = 0 [private]

Zähler der Knoten

Definiert in Zeile 43 der Datei `DependencyApplet.java`.

Wird benutzt von `DependencyApplet.init()`.

final Dimension DEFAULT_SIZE = new Dimension(1024, 480) [static, private]

Defaulte Größe des Applet

Definiert in Zeile 31 der Datei `DependencyApplet.java`.

Map<String, Bundle> idBundle = new HashMap<String, Bundle>() [private]

Map für Id und Bundle

Definiert in Zeile 48 der Datei `DependencyApplet.java`.

Wird benutzt von `DependencyApplet.init()`.

Bundle root [private]

Wurzel

Definiert in Zeile 53 der Datei `DependencyApplet.java`.

Wird benutzt von `DependencyApplet.getRoot()` und `DependencyApplet.init()`.

final long serialVersionUID = 1L [static, private]

Definiert in Zeile 33 der Datei `DependencyApplet.java`.

2.2.7 DependencyPanel Klassenreferenz

Abgeleitet von `javax.swing.JLayeredPane`.

2.2.7.1 Ausführliche Beschreibung

Diese Klasse `DependencyPanel` vererbt die `JLayeredPane` und wird zum `DependencyApplet` hinzugefügt. `DependencyPanel` legt fest, wie ein `BundleNode` auf einem Canvas lokalisiert und aufgemalt werden soll.

Autor:

Yu Liu

Definiert in Zeile 31 der Datei `DependencyPanel.java`.

Öffentliche Methoden

- `DependencyPanel (Bundle root)`
- `Point getCenter ()`
- `Bundle getRoot ()`
- `void paint (Graphics g)`
- `void setRoot (Bundle root)`
- `synchronized void startAnimation ()`
- `synchronized void stopAnimation ()`

Private Methoden

- `DependencyApplet getDependencyApplet ()`
- `void initializeGraph (Bundle parent, BundleNode parentNode, int row)`
- `void initializeNodeLocation ()`
- `boolean moveToPosition ()`

Private Attribute

- `Map< Bundle, BundleNode > bundleNode = new HashMap<Bundle, BundleNode>()`
- `int i = 0`
- `boolean initialized = false`

- `Map< Integer, Set< BundleNode > > levelNode = new HashMap<Integer, Set<BundleNode>>()`
- `int maxNumRow = 0`
- `Map< BundleNode, Point > nodeLocation = new HashMap<BundleNode, Point>()`
- Bundle root
- Timer timer

Statische private Attribute

- `static final Dimension BUTTON_SIZE = new Dimension(200, 25)`
- `static final long serialVersionUID = 1L`

Klassen

- `class TimerHandler`

2.2.7.2 Beschreibung der Konstruktoren und Destruktoren

DependencyPanel (Bundle root)

Konstruktor

Parameter:

root das Wurzelbundle

Definiert in Zeile 106 der Datei `DependencyPanel.java`.

Benutzt `DependencyPanel.setRoot()` und `DependencyPanel.timer`.

2.2.7.3 Dokumentation der Elementfunktionen

Point getCenter ()

das geometrisches Zentralpunkt

Rückgabe:

Definiert in Zeile 117 der Datei `DependencyPanel.java`.

Wird benutzt von `DependencyPanel.initializeNodeLocation()`.

DependencyApplet getDependencyApplet () [private]

Liefert `DependencyApplet` zurück.

Rückgabe:

Definiert in Zeile 226 der Datei `DependencyPanel.java`.

Wird benutzt von `DependencyPanel.initializeNodeLocation()` und `DependencyPanel.setRoot()`.

Bundle `getRoot ()`

Liefert das Wurzelbundle zurück.

Rückgabe:

Definiert in Zeile 127 der Datei `DependencyPanel.java`.

Benutzt `DependencyPanel.root`.

Wird benutzt von `BundleNode.BundleNodeMouseProcess.mouseClicked()`.

`void initializeGraph (Bundle parent, BundleNode parentNode, int row) [private]`

Initialisierung des Graphens

Parameter:

parent direkter Vorfahrbundle

parentNode direkter Vorfahrnode

row auf welcher ebene

Definiert in Zeile 242 der Datei `DependencyPanel.java`.

Benutzt `BundleNode.addConnection()`, `DependencyPanel.bundleNode`, `Bundle.getDependedBundles()`, `DependencyPanel.i`, `DependencyPanel.levelNode`, `DependencyPanel.maxNumRow` und `BundleNode.setBundle()`.

Wird benutzt von `DependencyPanel.setRoot()`.

`void initializeNodeLocation () [private]`

Initialisierung der Nodeposition

Definiert in Zeile 282 der Datei `DependencyPanel.java`.

Benutzt `DependencyPanel.getCenter()`, `DependencyPanel.getDependencyApplet()`, `DependencyPanel.levelNode`, `DependencyPanel.maxNumRow` und `DependencyPanel.nodeLocation`.

Wird benutzt von `DependencyPanel.paint()`.

boolean moveToPosition () [private]

Bewegt Nodes zu ihren Positionen.

Rückgabe:

Definiert in Zeile 331 der Datei DependencyPanel.java.

Benutzt DependencyPanel.i und DependencyPanel.nodeLocation.

void paint (Graphics g)

Diese Methode malt alle Bundlenode auf dem Canvas auf.

Definiert in Zeile 135 der Datei DependencyPanel.java.

Benutzt BundleNode.getCenter(), DependencyPanel.initialized, DependencyPanel.initializeNodeLocation(), BundleNode.paint() und DependencyPanel.startAnimation().

void setRoot (Bundle root)

Setter des Wurzelbundle

Parameter:**root**

Definiert in Zeile 173 der Datei DependencyPanel.java.

Benutzt DependencyPanel.bundleNode, DependencyPanel.getDependencyApplet(), Bundle.getName(), DependencyPanel.initialized, DependencyPanel.initializeGraph(), DependencyPanel.levelNode, DependencyPanel.maxNumRow, BundleNode.setBundle() und DependencyApplet.setDefaultSize().

Wird benutzt von DependencyPanel.DependencyPanel() und BundleNode.BundleNodeMouseProcess.mouseClicked().

synchronized void startAnimation ()

Startet die Animation.

Definiert in Zeile 207 der Datei DependencyPanel.java.

Benutzt DependencyPanel.timer.

Wird benutzt von DependencyPanel.paint().

synchronized void stopAnimation ()

Stopt die Animation.

Definiert in Zeile 216 der Datei DependencyPanel.java.

Benutzt DependencyPanel.timer.

2.2.7.4 Dokumentation der Datenelemente

Map<Bundle, BundleNode> bundleNode = new HashMap<Bundle, BundleNode>() [private]

Map für Bundle und BundleNode

Definiert in Zeile 62 der Datei DependencyPanel.java.

Wird benutzt von DependencyPanel.initializeGraph() und DependencyPanel.setRoot().

final Dimension BUTTON_SIZE = new Dimension(200, 25) [static, private]

Defaulte Größe eines BundleNode

Definiert in Zeile 55 der Datei DependencyPanel.java.

int i = 0 [private]

Zähler des BundleNode

Definiert in Zeile 67 der Datei DependencyPanel.java.

Wird benutzt von DependencyPanel.initializeGraph() und DependencyPanel.moveToPosition().

boolean initialized = false [private]

Initializationsflagge

Definiert in Zeile 72 der Datei DependencyPanel.java.

Wird benutzt von DependencyPanel.paint() und DependencyPanel.setRoot().

Map<Integer, Set<BundleNode> > levelNode = new HashMap<Integer, Set<BundleNode>>() [private]

In diesem Map werden Informationen gespeichert, die darauf zeigen, welche Node liegen auf welcher Ebene.

Definiert in Zeile 78 der Datei DependencyPanel.java.

Wird benutzt von DependencyPanel.initializeGraph(), DependencyPanel.initializeNodeLocation() und DependencyPanel.setRoot().

int maxNumRow = 0 [private]

Maximale Anzahl der Zeilen

Definiert in Zeile 83 der Datei DependencyPanel.java.

Wird benutzt von `DependencyPanel.initializeGraph()`,
`DependencyPanel.initializeNodeLocation()` und `DependencyPanel.setRoot()`.

Map<BundleNode, Point> nodeLocation = new HashMap<BundleNode, Point>() [private]

In diesem Map werden BundleNodes und ihre Position gespeichert.

Definiert in Zeile 88 der Datei `DependencyPanel.java`.

Wird benutzt von `DependencyPanel.initializeNodeLocation()` und
`DependencyPanel.moveToPosition()`.

Bundle root [private]

die Wurzel

Definiert in Zeile 93 der Datei `DependencyPanel.java`.

Wird benutzt von `DependencyPanel.getRoot()`.

final long serialVersionUID = 1L [static, private]

Definiert in Zeile 57 der Datei `DependencyPanel.java`.

Timer timer [private]

der Timer der Animation

Definiert in Zeile 98 der Datei `DependencyPanel.java`.

Wird benutzt von `DependencyPanel.DependencyPanel()`,
`DependencyPanel.startAnimation()` und `DependencyPanel.stopAnimation()`.

2.2.8 DependencyPanel.TimerHandler Klassenreferenz

Abgeleitet von `java::awt::event::ActionListener`.

2.2.8.1 Ausführliche Beschreibung

Diese Klasse `TimerHandler` wird zur Animation benötigt.

Autor:

Yu

Definiert in Zeile 39 der Datei `DependencyPanel.java`.

Öffentliche Methoden

- void actionPerformed (ActionEvent arg0)

2.2.8.2 Dokumentation der Elementfunktionen

void actionPerformed (ActionEvent arg0)

Definiert in Zeile 42 der Datei DependencyPanel.java.

2.2.9 GraphicsUtils Klassenreferenz

2.2.9.1 Ausführliche Beschreibung

Diese Klasse GraphicsUtils ist eine Werkzeugklasse, die geometrische Methoden anbietet, damit das Aufmalen der Bundlenode erleichtert wird.

Autor:

Yu Liu

Definiert in Zeile 14 der Datei GraphicsUtils.java.

Öffentliche, statische Methoden

- static Rectangle createRectangle (Point point1, Point point2)
- static Rectangle createRectangle (BundleNode n1, BundleNode n2)
- static Point[] getClosestAnchoragePoint (Point[] point1, Point[] point2)
- static Point[] getClosestAnchoragePoint (BundleNode n1, BundleNode n2)
- static int[] getClosestSide (BundleNode n1, BundleNode n2)
- static Point[] getContainedPoints (BundleNode n, Rectangle r)
- static int getSide (BundleNode node, Point point)

2.2.9.2 Dokumentation der Elementfunktionen

static Rectangle createRectangle (Point point1, Point point2) [static]

Erzeugen eines Rechteck zwischen zwei Punkten

Parameter:

point1 der erste Punkt

point2 der zweite Punkt

Rückgabe:

ein Rechteck zwischen zwei Punkten

Definiert in Zeile 47 der Datei GraphicsUtils.java.

static Rectangle createRectangle (BundleNode n1, BundleNode n2) [static]

Erzeugen eines Rechteck zwischen zwei Bundlenodes

Parameter:

n1 das erste Node

n2 das zweite Node

Rückgabe:

ein Rechteck zwischen zwei Nodes

Definiert in Zeile 24 der Datei GraphicsUtils.java.

static Point [] getClosestAnchoragePoint (Point[] point1, Point[] point2) [static]

Liefert die nächsten Ankerungspunkte zwischen zwei Punkten

Parameter:

point1 der erste Punkte

point2 der zweite Punkte

Rückgabe:

die nächsten Punkte

Definiert in Zeile 111 der Datei GraphicsUtils.java.

static Point [] getClosestAnchoragePoint (BundleNode n1, BundleNode n2) [static]

Die nächsten Ankerungspunkte zwischen zwei BundleNodes

Parameter:

n1 das erste BundleNode

n2 das zweite BundleNode

Rückgabe:

die Ankerungspunkte

Definiert in Zeile 89 der Datei GraphicsUtils.java.

static int [] getClosestSide (BundleNode n1, BundleNode n2) [static]

Die nächsten Seiten zwischen zwei Bundlenodes

Parameter:

n1 das erste BundleNode

n2 das zweite BundleNode

Rückgabe:

die kürzeste Seiten

Definiert in Zeile 66 der Datei GraphicsUtils.java.

static Point [] getContainedPoints (BundleNode n, Rectangle r) [static]

Liefert die Punkte eines BundleNode zurück, die von einem Recheck enthalten wird.

Parameter:

n das BundleNode

r das Recheck

Rückgabe:

die enthaltene Punkte

Definiert in Zeile 144 der Datei GraphicsUtils.java.

Benutzt BundleNode.getPartPoint().

static int getSide (BundleNode node, Point point) [static]

Liefert Seiten zurück, in deren Nähe sich ein Point befindet.

Parameter:

node das BundleNode

point das Point

Rückgabe:

die Seite

Definiert in Zeile 185 der Datei GraphicsUtils.java.

2.2.10 BundleDependencyUtils Klassenreferenz

2.2.10.1 Ausführliche Beschreibung

BundleDependencyUtils ist eine Werkzeug-Klasse, die eine Analyse der Bundle-Abhängigkeiten erleichtert. Es fehlt in OSGI direkte Methoden, die die abhängigen Bundles eines Bundles herausfinden können und ebenso die von einem Bundle abhängigen Bundles. Solche Methoden sind grundlegende Bausteine der Abhängigkeitsanalyse der Bundles. Dafür soll diese Werkzeug-Klasse hier eine gute Hilfe geben.

Autor:

Yu Liu

Definiert in Zeile 26 der Datei BundleDependencyUtils.java.

Öffentliche, statische Methoden

- static void getAllRequiringBundles (long bundleId, Set< Bundle > requiringBundles)
- static Bundle[] getDependedBundles (long id)
- static Bundle[] getDependedBundles (Bundle bundle)
- static ExportedPackage[] getExportedPackages (Bundle bundle)
- static Bundle getExportingBundle (ExportedPackage exportedPackage)
- static Bundle[] getImportingBundles (ExportedPackage exportedPackage)
- static RequiredBundle[] getRequiredBundles (String symbolicName)
- static Bundle[] getRequiringBundles (RequiredBundle requiredBundle)

Private, statische Methoden

- static PackageAdmin getPackageAdmin ()

2.2.10.2 Dokumentation der Elementfunktionen**static void getAllRequiringBundles (long bundleId, Set< Bundle > requiringBundles)**
[static]

Liefert Bundles zurück, die ein übergebenes Bundle direkt oder indirekt benötigen.

Siehe auch:

getRequiringBundles(RequiredBundle)

Parameter:**bundleId** Bundle-Id**requiringBundles** direkt oder indirekt benötigen Bundles

Definiert in Zeile 160 der Datei BundleDependencyUtils.java.

static Bundle [] getDependedBundles (long id) [static]

Liefert Bundles zurück, davon ein übergebenes Bundle abhängig ist. Das Bundle wird durch eine übergebene Bundle-Id repräsentiert.

Siehe auch:

getDependedBundles(Bundle)

Parameter:

id Bundle-Id

Rückgabe:

abgehangene Bundles

Definiert in Zeile 75 der Datei BundleDependencyUtils.java.

static Bundle [] getDependedBundles (Bundle bundle) [static]

Liefert Bundles zurück, davon ein übergebenes **Bundle** abhängig ist.

Siehe auch:

getDependedBundles(long)

Parameter:

bundle das übergegebene Bundle

Rückgabe:

abgehangene Bundles

Definiert in Zeile 38 der Datei BundleDependencyUtils.java.

static ExportedPackage [] getExportedPackages (Bundle bundle) [static]

Liefert Packages zurück, die ein übergebenes **Bundle** exportiert.

Parameter:

bundle das übergegebene Bundle

Rückgabe:

exportierte Packages

Definiert in Zeile 88 der Datei BundleDependencyUtils.java.

static Bundle getExportingBundle (ExportedPackage exportedPackage) [static]

Liefert ein **Bundle** zurück, das ein übergebenes Package exportiert.

Parameter:

exportedPackage das übergegebene Package

Rückgabe:

ein exportierende **Bundle**

Definiert in Zeile 104 der Datei BundleDependencyUtils.java.

static Bundle [] getImportingBundles (ExportedPackage exportedPackage) [static]

Liefert Bundles zurück, das ein übergegebenes Package importieren

Parameter:

exportedPackage das übergebene Package

Rückgabe:

importierende Bundles

Definiert in Zeile 115 der Datei BundleDependencyUtils.java.

static PackageAdmin getPackageAdmin () [static, private]

Liefert PackageAdmin zurück, die den Zugriff auf installierte Bundles ermöglicht.

Rückgabe:

PackageAdmin

Definiert in Zeile 190 der Datei BundleDependencyUtils.java.

static RequiredBundle [] getRequiredBundles (String symbolicName) [static]

Liefert Bundles zurück, die einen übergebenen Namen haben.

Parameter:

symbolicName der übergebene Name

Rückgabe:

benötigte Bundles

Definiert in Zeile 127 der Datei BundleDependencyUtils.java.

static Bundle [] getRequiringBundles (RequiredBundle requiredBundle) [static]

Liefert Bundles zurück, die ein übergegebenes Bundle benötigen, und zwar importieren.

Siehe auch:

getAllRequiringBundles()

Parameter:

requiredBundle das übergebene Bundle

Rückgabe:

importierende Bundles

Definiert in Zeile 145 der Datei BundleDependencyUtils.java.

2.2.11 BundleUsers Klassenreferenz

2.2.11.1 Ausführliche Beschreibung

Die Objekte der Klasse BundleUsers werden als Bean der Nutzeranteil-Statistik der EAM-Bundles benötigt. Die Statistik zeigt, wieviele Nutzer arbeiten an einem Bundle und auch den Nutzersanteil eines Bundle.

Autor:

Yu Liu

Definiert in Zeile 31 der Datei BundleUsers.java.

Öffentliche Methoden

- List< User > getBundleUsers ()
- String getContent ()
- ArrayList< Bundle > getEamBundles ()
- String getSortColumnOver ()
- String getTitle ()
- boolean isSortAscendingOver ()
- boolean isToggleEditMessage ()
- boolean isToggleSelectAll ()
- void notificate ()
- void selectBundleUsers ()
- void setBundleUsers (List< User > bundleUsers)
- void setContent (String content)
- void setSortAscendingOver (boolean sortAscendingOver)
- void setSortColumnOver (String sortColumnOver)
- void setTitle (String title)
- void setToggleEditMessage (boolean toggleEditMessage)
- void setToggleSelectAll (boolean toggleSelectAll)
- void toggleEditMessage ()
- void toggleSelectAll ()

Private Attribute

- List< User > bundleUsers = new LinkedList<User>()
- String content = ""
- boolean sortAscendingOver = true
- String sortColumnOver = null

- `String titel = ""`
- `boolean toggleEditMessage = false`
- `boolean toggleSelectAll = false`

2.2.11.2 Dokumentation der Elementfunktionen

List<User> getBundleUsers ()

Liefert Nutzer zurück, die gerade ein Bundle benutzen.

Rückgabe:

ein `List` von `User`

Definiert in Zeile 73 der Datei `BundleUsers.java`.

Benutzt `BundleUsers.bundleUsers`.

String getContent ()

Liefert den Inhalt einer Systemnachricht zurück, die gesendet wird.

Rückgabe:

der Inhalt einer Systemnachricht

Definiert in Zeile 82 der Datei `BundleUsers.java`.

Benutzt `BundleUsers.content`.

ArrayList<Bundle> getEamBundles ()

Liefert alle Bundles des EAM-Systems zurück.

Rückgabe:

ein `ArrayList` von `Bundle`

Definiert in Zeile 91 der Datei `BundleUsers.java`.

String getSortColumnOver ()

Liefert den Namen eines Columns zurück, danach die Nutzertabelle sortiert wird.

Rückgabe:

der Name eines Columns

Definiert in Zeile 113 der Datei `BundleUsers.java`.

Benutzt `BundleUsers.sortColumnOver`.

String getTitle ()

Liefert den Titel einer Systemnachricht zurück.

Rückgabe:

der Titel einer Systemnachricht

Definiert in Zeile 122 der Datei BundleUsers.java.

Benutzt BundleUsers.titel.

boolean isSortAscendingOver ()

Prüft, ob die Bundlesnutzer aufsteigend sortiert ist.

Rückgabe:

true, falls die Tabelle aufsteigend sortiert ist. **false**, falls die Tabelle absteigend sortiert ist.

Definiert in Zeile 132 der Datei BundleUsers.java.

Benutzt BundleUsers.sortAscendingOver.

boolean isToggleEditMessage ()

Prüft, ob der Edit-Message-Button getriggert ist.

Rückgabe:

true, falls der Edit-Message-Button getriggert ist. **false**, falls der Edit-Message-Button nicht getriggert ist.

Definiert in Zeile 143 der Datei BundleUsers.java.

Benutzt BundleUsers.toggleEditMessage().

boolean isToggleSelectAll ()

Prüft, ob der Select-All-Button getirggert ist.

Rückgabe:

true, falls der Select-All-Button getriggert ist. **false**, falls der Select-All-Button nicht getriggert ist.

Definiert in Zeile 154 der Datei BundleUsers.java.

Benutzt BundleUsers.toggleSelectAll().

void notificate ()

Diese Methode schickt eine editierte Nachricht an die gewählten Nutzer, wenn der Send-Button geklickt wird.

Definiert in Zeile 162 der Datei BundleUsers.java.

Benutzt BundleUsers.bundleUsers, BundleUsers.content, BundleUsers.titel und BundleUsers.toggleEditMessage().

void selectBundleUsers ()

Diese Methode findet alle direkten und indirekten Nutzer heraus, die ein selektiertes Bundle benutzen.

Definiert in Zeile 176 der Datei BundleUsers.java.

Benutzt BundleUsers.bundleUsers.

void setBundleUsers (List< User > bundleUsers)

setter des NutzerLists eines Bundle

Parameter:

bundleUsers ein neues List der Nutzer eines Bundle

Definiert in Zeile 244 der Datei BundleUsers.java.

void setContent (String content)

setter des Inhalts einer Systemnachricht

Parameter:

content der neue Inhalt einer Systemnachricht

Definiert in Zeile 254 der Datei BundleUsers.java.

void setSortAscendingOver (boolean sortAscendingOver)

setter der Flagge für die Sortierung

Parameter:

sortAscendingOver aufsteigend(**true**) oder absteigend(**false**)

Definiert in Zeile 264 der Datei BundleUsers.java.

void setSortColumnOver (String sortColumnOver)

setter des Column-Namens, wonach die Tabelle sortiert wird.

Parameter:

sortColumnOver der neue Column-Name

Definiert in Zeile 274 der Datei BundleUsers.java.

void setTitle (String title)

setter des Titels einer Nachricht

Parameter:

title der Titel

Definiert in Zeile 284 der Datei BundleUsers.java.

Benutzt BundleUsers.titel.

void setToggleEditMessage (boolean toggleEditMessage)

setter der Triggerflagge des Edit-Message-Buttons

Parameter:

toggleEditMessage true gesetzt, false nicht gesetzt.

Definiert in Zeile 295 der Datei BundleUsers.java.

void setToggleSelectAll (boolean toggleSelectAll)

setter der Triggerflagge des Select-All-Buttons

Parameter:

toggleSelectAll true gesetzt, false nicht gesetzt.

Definiert in Zeile 306 der Datei BundleUsers.java.

void toggleEditMessage ()

Edit-Message-Button triggert

Definiert in Zeile 313 der Datei BundleUsers.java.

Wird benutzt von BundleUsers.isToggleEditMessage() und BundleUsers.notify().

void toggleSelectAll ()

Select-All-Button triggert

Definiert in Zeile 320 der Datei BundleUsers.java.

Wird benutzt von BundleUsers.isToggleSelectAll().

2.2.11.3 Dokumentation der Datenelemente

List<User> bundleUsers = new LinkedList<User>() [private]

ein `List` von Nutzern eines Bundle

Definiert in Zeile 36 der Datei `BundleUsers.java`.

Wird benutzt von `BundleUsers.getBundleUsers()`, `BundleUsers.notify()` und `BundleUsers.selectBundleUsers()`.

`String content = ""` [private]

der Inhalt einer editierten Nachricht

Definiert in Zeile 41 der Datei `BundleUsers.java`.

Wird benutzt von `BundleUsers.getContent()` und `BundleUsers.notify()`.

`boolean sortAscendingOver = true` [private]

die Triggerflagge der Sortierung

Definiert in Zeile 46 der Datei `BundleUsers.java`.

Wird benutzt von `BundleUsers.isSortAscendingOver()`.

`String sortColumnOver = null` [private]

der Column-Name, danach die Tabelle sortiert ist.

Definiert in Zeile 51 der Datei `BundleUsers.java`.

Wird benutzt von `BundleUsers.getSortColumnOver()`.

`String titel = ""` [private]

der Titel einer editierten Nachricht

Definiert in Zeile 56 der Datei `BundleUsers.java`.

Wird benutzt von `BundleUsers.getTitle()`, `BundleUsers.notify()` und `BundleUsers.setTitle()`.

`boolean toggleEditMessage = false` [private]

die Triggerflagge des Edit-Message-Buttons

Definiert in Zeile 61 der Datei `BundleUsers.java`.

`boolean toggleSelectAll = false` [private]

die Triggerflagge des Select-All-Buttons

Definiert in Zeile 66 der Datei `BundleUsers.java`.

2.2.12 ImageShow Klassenreferenz

2.2.12.1 Ausführliche Beschreibung

Die Klasse `ImageShow` generiert den Pizza-Chart in der PNG-Format für die Nutzeranteil-Statistik. `ImageShow` benutzt im Hintergrund die `JFreechart`-Bibliotheken, die eine sehr bekannte Open-Source-Bibliothek zur Generierung verschiedener Statistik-Charts ist. Diese Bibliothek kann sowohl in Desktop-Java-Anwendungen als auch in Web-Applikationen verwendet werden. Für Web-Applikationen bittet diese Bibliothek eine Klasse `ServletUtilities` an. Diese Klasse ermöglicht den Web-Applikationen, die generierten Bilder mit Hilfe der Klasse `DisplayChartServlet` zuzugreifen, das man vorher mit Hilfe `HttpService` in OSGI registrieren kann.

Autor:

Yu Liu

Definiert in Zeile 36 der Datei `ImageShow.java`.

Öffentliche Methoden

- `String getImageurl (HttpServletRequest request, HttpServletResponse response)`

Private Methoden

- `DefaultPieDataset getdataset (HttpServletRequest request)`

2.2.12.2 Dokumentation der Elementfunktionen

DefaultPieDataset getdataset (HttpServletRequest request) [private]

Liefert verarbeiteten Daten zurück, die die `JFreeChart`-Bibliothek zur Generierung eines Charts benutzen kann. Diese Methode liest zuerst die Parameter aus einem `HttpServletRequest` und ermittelt dann mit Hilfe der Klasse `PageAccessTrackServlet`, wieviele Nutzer an einem Bundle arbeiten und schließlich packt solche Informationen als `DefaultPieDataset` zusammen, die als Eingabe des Generierungsalgorithmus der Charts angesehen werden könnte.

Parameter:

request ein `HttpServletRequest`

Rückgabe:

Chart-Daten

Definiert in Zeile 93 der Datei `ImageShow.java`.

Wird benutzt von `ImageShow.getImageurl()`.

String getImageurl (HttpServletRequest request, HttpServletResponse response)

Liefert einen URL zurück, das auf den generierten Bild verweist. In dieser Methode wird die Klasse `ServletUtilities` verwendet, um den Chart zu generieren.

Parameter:

request ein `HttpServletRequest`

response ein `HttpServletResponse`

Rückgabe:

ein URL des generierten Bilds(Chart)

Definiert in Zeile 49 der Datei `ImageShow.java`.

Benutzt `ImageShow.getdataset()`.

2.2.13 ParamsGenerator Klassenreferenz**2.2.13.1 Ausführliche Beschreibung**

Clientseitig werden die Ergebnisse der Abhängigkeitsanalyse der Bundles mit Hilfe eines Applets dargestellt. Zur Steuerung der Darstellung generiert `ParamsGenerator` die Analysedaten als Applet-Parameters, die später an das Applet übergegeben werden. Das Applet liest die Daten ein und stellt die Abhängigkeiten der Bundles in einer entsprechenden Baumstruktur dar. Die Applet-Parameters sind eine Gruppe der Wertpaare, die sich aus einem Namen und einem Value zusammensetzen. Der Name enthält die Informationen, die darauf hinweisen, auf welcher Ebene eines Baums sich ein Node befindet und wer sein direkter Vorfahr ist, während der Value seine Abhängigkeiten mit anderen Nodes angibt.

Autor:

Yu Liu

Definiert in Zeile 27 der Datei `ParamsGenerator.java`.

Öffentliche Methoden

- `Map< String, String > getDependedBundle (long bundleId)`

Statische öffentliche Attribute

- `static final String LINKED_BY_PARAM = "LINKED_BY"`
- `static final String NAME_PREFIX = "LEVEL"`
- `static final String SYMBOLIC_NAME_PARAM = "SYMBOLIC_NAME"`

Private Methoden

- void getDependedBundles (Bundle parent, int level)

Private Attribute

- Map< Bundle, String > bundleName = new HashMap<Bundle, String>()
- Set< Bundle > dependedBundles = new HashSet<Bundle>()
- Map< String, String > params = new HashMap<String, String>()

2.2.13.2 Dokumentation der Elementfunktionen

Map<String, String> getDependedBundle (long bundleId)

Liefert ein Map zurück, das die Analysedaten als Wertpaare(Name/Value) repräsentiert. Das Algorithmus hier wird die Analysedaten rekursiv verarbeitet und dabei aufpassen, das die Abhängigkeiten zwischen Bundles nicht nur transitiv sondern auch zyklisch sein könnten. Dewegen ist die Kondition, ob das rekursive Algorithmus enden soll, wird mit Hilfe der Mengenoperationen getestet.

Parameter:

bundleId Bundle-Id

Rückgabe:

ein Map, das Wertpaare(Name/Value) enthält.

Definiert in Zeile 75 der Datei ParamsGenerator.java.

Benutzt ParamsGenerator.bundleName, ParamsGenerator.getDependedBundles() und ParamsGenerator.params.

Wird benutzt von BundleManager.getDetails().

void getDependedBundles (Bundle parent, int level) [private]

Diese Methode ist eine interne Hilfe der Methode getDependedBundle(long) beim rekursiven Erzeugen der Applet-Parameter

Parameter:

parent der Vater-Node

level gibt an, auf welcher Ebene befindet sich der Vater-Node.

Definiert in Zeile 100 der Datei ParamsGenerator.java.

Benutzt ParamsGenerator.bundleName, ParamsGenerator.dependedBundles und ParamsGenerator.params.

Wird benutzt von ParamsGenerator.getDependedBundle().

2.2.13.3 Dokumentation der Datenelemente

Map<Bundle, String> bundleName = new HashMap<Bundle, String>() [private]

Ein Map für Bundle und seinen symbolischen Namen

Definiert in Zeile 51 der Datei ParamsGenerator.java.

Wird benutzt von ParamsGenerator.getDependedBundle() und ParamsGenerator.getDependedBundles().

Set<Bundle> dependedBundles = new HashSet<Bundle>() [private]

Eine Menge der abhängigen Bundles

Definiert in Zeile 56 der Datei ParamsGenerator.java.

Wird benutzt von ParamsGenerator.getDependedBundles().

final String LINKED_BY_PARAM = "LINKED_BY" [static]

der Schlüssel eines Wertpaars der Applet-Parameter. Sein Wert gibt an, mit welchem Node ein Node die Abhängigkeit hat.

Definiert in Zeile 33 der Datei ParamsGenerator.java.

final String NAME_PREFIX = "LEVEL" [static]

der Schlüssel eines Wertpaars der Applet-Parameter. Sein Wert gibt an, auf welcher Ebene befindet sich ein Node.

Definiert in Zeile 39 der Datei ParamsGenerator.java.

Map<String, String> params = new HashMap<String, String>() [private]

Ein Map der Applet-Parameter

Definiert in Zeile 61 der Datei ParamsGenerator.java.

Wird benutzt von ParamsGenerator.getDependedBundle() und ParamsGenerator.getDependedBundles().

final String SYMBOLIC_NAME_PARAM = "SYMBOLIC_NAME" [static]

der Schlüssel eines Wertpaars der Applet-Parameter. Sein Wert ist der symbolischen Name eines Nodes.

Definiert in Zeile 45 der Datei ParamsGenerator.java.

2.3 Matchinghandler

2.3.1 BundleConfigurationNavigation Klassenreferenz

2.3.1.1 Ausführliche Beschreibung

Diese Klasse dient der Navigation zwischen den verschiedenen Seiten der Bundle-Konfiguration. Dazu wird immer die aktuelle Seite zwischengespeichert und davon abhängig die boolean-Flags auf der letzten Seite auf false zurückgesetzt.

Zusätzlich speichert die Klasse die Equinox-Id des Bundles, um von jeder Seite auf diese zugreifen zu können.

Autor:

Jens Henkel

Definiert in Zeile 18 der Datei BundleConfigurationNavigation.java.

Öffentliche Methoden

- BundleConfigurationNavigation ()
- String goToDependencies ()
- String goToObjectAttributes ()
- String goToObjectMappingPage ()
- String goToObjects ()
- String goToObjectStoringPage ()
- String goToOverview ()
- String goToRelationAttributes ()
- String goToRelationMappingPage ()
- String goToRelations ()
- String goToRelationStoringPage ()
- String goToRoles ()
- boolean isBundleHasObjects ()
- boolean isBundleHasRelations ()
- boolean isBundleHasViews ()
- boolean isRolesActive ()

Öffentliche, statische Methoden

- static String getActualPage ()
- static String getBundleDbId ()
- static String getBundleEquinoxId ()

- static BundleConfigurationNavigation getInstance ()
- static void setActualPage (String actualPage)
- static void setBundleDbId (String bundleDbId)
- static void setBundleEquinoxId (String bundleEquinoxId)
- static void setBundleHasObjects (boolean bundleHasObjects)
- static void setBundleHasRelations (boolean bundleHasRelations)
- static void setBundleHasViews (boolean bundleHasViews)

Statische private Attribute

- static String actualPage
- static String bundleDbId
- static String bundleEquinoxId
- static boolean bundleHasObjects = false
- static boolean bundleHasRelations = false
- static boolean bundleHasViews = false
- static BundleConfigurationNavigation instance

2.3.1.2 Beschreibung der Konstruktoren und Destruktoren

BundleConfigurationNavigation ()

Singleton-Konstruktor

Definiert in Zeile 46 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.instance.

Wird benutzt von BundleConfigurationNavigation.getInstance().

2.3.1.3 Dokumentation der Elementfunktionen

static String getActualPage () [static]

Gibt die aktuelle Seite als String zurück.

Rückgabe:

Die aktuelle Seite als String.

Definiert in Zeile 135 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.actualPage.

static String getBundleDbId () [static]

Definiert in Zeile 33 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.bundleDbId.

static String getBundleEquinoxId () [static]

Definiert in Zeile 64 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.bundleEquinoxId.

static BundleConfigurationNavigation getInstance () [static]

Rückgabe:

der verweist auf das Singleton-Objekt dieser Klasse.

Definiert in Zeile 55 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.BundleConfigurationNavigation() und BundleConfigurationNavigation.instance.

String goToDependencies ()

Springt zu der Seite der Views und deren Abhängigkeiten.

Rückgabe:

Der String der Abhängigkeiten-Seite ("dependencies").

Definiert in Zeile 428 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.actualPage.

String goToObjectAttributes ()

Springt zu der Seite des Objekt-Attribut-Mappings und setzt die jeweiligen Boolean-Flags zurück.

Rückgabe:

Der String der Objekt-Attribut-Mapping-Seite ("objectAttributes").

Definiert in Zeile 286 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.actualPage.

String goToObjectMappingPage ()

Springt zu der Seite der Objektverknüpfungen, ruft goToObjects auf.

Rückgabe:

Liefert den String "objects" zurück.

Definiert in Zeile 478 der Datei BundleConfigurationNavigation.java.

Benutzt `BundleConfigurationNavigation.actualPage` und `BundleConfigurationNavigation.goToObjects()`.

String goToObjects ()

Springt zu der Seite des Objekt-Mappings und setzt die jeweiligen Boolean-Flags zurück.

Rückgabe:

Der String der Objekt-Mapping-Seite ("objects").

Definiert in Zeile 155 der Datei `BundleConfigurationNavigation.java`.

Benutzt `BundleConfigurationNavigation.actualPage`.

Wird benutzt von `BundleConfigurationNavigation.goToObjectMappingPage()` und `BundleConfigurationNavigation.goToObjectStoringPage()`.

String goToObjectStoringPage ()

Springt zu der Seite der Objektspeicherung, ruft `goToObjects` auf.

Rückgabe:

Liefert den String "objects" zurück.

Definiert in Zeile 498 der Datei `BundleConfigurationNavigation.java`.

Benutzt `BundleConfigurationNavigation.actualPage` und `BundleConfigurationNavigation.goToObjects()`.

String goToOverview ()

Springt zur Bundle-Overview-Seite zurück und setzt alle Boolean-Flags zurück.

Rückgabe:

"overview"

Definiert in Zeile 331 der Datei `BundleConfigurationNavigation.java`.

String goToRelationAttributes ()

Springt zu der Seite des Relationen-Attribut-Mappings und setzt die jeweiligen Boolean-Flags zurück.

Rückgabe:

Der String der Relation-Attribut-Seite ("relationAttributes").

Definiert in Zeile 379 der Datei `BundleConfigurationNavigation.java`.

Benutzt `BundleConfigurationNavigation.actualPage`.

String goToRelationMappingPage ()

Springt zu der Seite der Relationsverknüpfungen, ruft goToRelations auf.

Rückgabe:

Liefert den String "relations" zurück.

Definiert in Zeile 518 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.actualPage und BundleConfigurationNavigation.goToRelations().

String goToRelations ()

Springt zu der Seite des Relationen-Mappings und setzt die jeweiligen Boolean-Flags zurück.

Rückgabe:

Der String der Relationen-Mapping-Seite ("relations").

Definiert in Zeile 241 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.actualPage.

Wird benutzt von BundleConfigurationNavigation.goToRelationMappingPage() und BundleConfigurationNavigation.goToRelationStoringPage().

String goToRelationStoringPage ()

Springt zu der Seite der Relationsspeicherung, ruft goToRelations auf.

Rückgabe:

Liefert den String "relations" zurück.

Definiert in Zeile 538 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.actualPage und BundleConfigurationNavigation.goToRelations().

String goToRoles ()

Springt zu der Seite des Rollen-Mappings und setzt die jeweiligen Boolean-Flags zurück.

Rückgabe:

Der String der Rollen-Mapping-Seite ("roles").

Definiert in Zeile 200 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.actualPage.

boolean isBundleHasObjects ()

Methode, die für das Anzeigen des Menus in navigation.jspf verwendet wird.

Rückgabe:

true, wenn das Bundle über EAM-Objekte verfügt, false sonst.

Definiert in Zeile 105 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.bundleHasObjects.

boolean isBundleHasRelations ()

Methode, die für das Anzeigen des Menus in navigation.jspf verwendet wird.

Rückgabe:

true, wenn das Bundle über EAM-Relationen verfügt, false sonst.

Definiert in Zeile 120 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.bundleHasRelations.

boolean isBundleHasViews ()

Methode, die für das Anzeigen des Menus in navigation.jspf verwendet wird.

Rückgabe:

true, wenn das Bundle über Rollen verfügt, false sonst.

Definiert in Zeile 90 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.bundleHasViews.

boolean isRolesActive ()

Definiert in Zeile 75 der Datei BundleConfigurationNavigation.java.

Benutzt BundleConfigurationNavigation.bundleHasViews.

static void setActualPage (String actualPage) [static]

Setzt die Seite, auf der sich die konfiguration derzeit befindet.

Parameter:

actualPage Die Seite, auf der sich die Konfiguration derzeit befindet.

Definiert in Zeile 145 der Datei BundleConfigurationNavigation.java.

static void setBundleDbId (String bundleDbId) [static]

Definiert in Zeile 38 der Datei BundleConfigurationNavigation.java.

static void setBundleEquinoxId (String bundleEquinoxId) [static]

Definiert in Zeile 69 der Datei BundleConfigurationNavigation.java.

static void setBundleHasObjects (boolean bundleHasObjects) [static]

Definiert in Zeile 110 der Datei BundleConfigurationNavigation.java.

static void setBundleHasRelations (boolean bundleHasRelations) [static]

Definiert in Zeile 125 der Datei BundleConfigurationNavigation.java.

static void setBundleHasViews (boolean bundleHasViews) [static]

Definiert in Zeile 95 der Datei BundleConfigurationNavigation.java.

2.3.1.4 Dokumentation der Datenelemente

String actualPage [static, private]

Definiert in Zeile 20 der Datei BundleConfigurationNavigation.java.

Wird benutzt von BundleConfigurationNavigation.getActualPage(), BundleConfigurationNavigation.goToDependencies(), BundleConfigurationNavigation.goToObjectAttributes(), BundleConfigurationNavigation.goToObjectMappingPage(), BundleConfigurationNavigation.goToObjectStoringPage(), BundleConfigurationNavigation.goToRelationAttributes(), BundleConfigurationNavigation.goToRelationMappingPage(), BundleConfigurationNavigation.goToRelations(), BundleConfigurationNavigation.goToRelationStoringPage() und BundleConfigurationNavigation.goToRoles().

String bundleDbId [static, private]

Definiert in Zeile 30 der Datei BundleConfigurationNavigation.java.

Wird benutzt von BundleConfigurationNavigation.getBundleDbId().

String bundleEquinoxId [static, private]

Definiert in Zeile 28 der Datei BundleConfigurationNavigation.java.

Wird benutzt von BundleConfigurationNavigation.getBundleEquinoxId().

boolean bundleHasObjects = false [static, private]

Definiert in Zeile 25 der Datei BundleConfigurationNavigation.java.

Wird benutzt von BundleConfigurationNavigation.isBundleHasObjects().

boolean bundleHasRelations = false [static, private]

Definiert in Zeile 26 der Datei BundleConfigurationNavigation.java.

Wird benutzt von BundleConfigurationNavigation.isBundleHasRelations().

boolean bundleHasViews = false [static, private]

Definiert in Zeile 24 der Datei BundleConfigurationNavigation.java.

Wird benutzt von BundleConfigurationNavigation.isBundleHasViews() und BundleConfigurationNavigation.isRolesActive().

BundleConfigurationNavigation instance [static, private]

Definiert in Zeile 22 der Datei BundleConfigurationNavigation.java.

Wird benutzt von BundleConfigurationNavigation.BundleConfigurationNavigation() und BundleConfigurationNavigation.getInstance().

2.3.2 CoreBundleObjectAttributeMatchingHandler Klassenreferenz

2.3.2.1 Ausführliche Beschreibung

ManagedBean für das Mapping der Objekt-Attribute des Objektes des Kerns und des Objektes des Bundles sowie das Anfügen von Attributen an ein Kernobjekt.

Enthält gesamte Logik, die für diese Aufgaben benötigt werden.

Autor:

Jens Henkel

Definiert in Zeile 28 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Öffentliche Methoden

- void checkDelete ()
- void checkDeleteMarkedElements ()
- CoreBundleObjectAttributeMatchingHandler ()
- void deleteAssignment ()
- void deleteSelectedAssignments ()
- ArrayList< EAMObjectAttributeMatchingStore > getAttributeMatches ()
- EAMObject getBundleObject ()

- `ArrayList< SelectItem > getBundleObjectAttributeSelectItems ()`
- `EAMObject getCoreObject ()`
- `ArrayList< SelectItem > getCoreObjectAttributeSelectItems ()`
- `ArrayList< String > getSelectedBundleAttributes ()`
- `String getSelectedBundleObjectAttribute ()`
- `String getSelectedCoreObjectAttribute ()`
- `void goToMappingPage ()`
- `void goToStoringPage ()`
- `boolean isAttributeMappingPage ()`
- `boolean isAttributeMatchesEmpty ()`
- `boolean isAttributeStoringPage ()`
- `boolean isDeleteMappedAndStored ()`
- `boolean isDeleteMappedAttribute ()`
- `boolean isDeleteMappedAttributeSelected ()`
- `boolean isDeleteStoredAttribute ()`
- `boolean isDeleteStoredAttributeSelected ()`
- `boolean isSizeGreaterTen ()`
- `void noDelete ()`
- `void resetFlags ()`
- `void saveAttributeAssignment ()`
- `void setDeleteMappedAndStored (boolean deleteMappedAndStored)`
- `void setDeleteMappedAttribute (boolean deleteMappedAttribute)`
- `void setDeleteMappedAttributeSelected (boolean deleteMappedAttributeSelected)`
- `void setDeleteStoredAttribute (boolean deleteStoredAttribute)`
- `void setDeleteStoredAttributeSelected (boolean deleteStoredAttributeSelected)`
- `void setSelectedBundleAttributes (ArrayList< String > selectedBundleAttributes)`
- `void setSelectedBundleObjectAttribute (String selectedBundleObjectAttribute)`
- `void setSelectedCoreObjectAttribute (String selectedCoreObjectAttribute)`
- `void storeAttributesToObject ()`

Öffentliche, statische Methoden

- `static boolean isAttributeMatchesGenerated ()`
- `static boolean isBundleSelectItemsGenerated ()`
- `static boolean isCoreSelectItemsGenerated ()`
- `static void setAttributeMappingPage (boolean attributeMappingPage)`
- `static void setAttributeMatchesEmpty (boolean attributeMatchesEmpty)`
- `static void setAttributeMatchesGenerated (boolean attributeMatchesGenerated)`

- static void setAttributeStoringPage (boolean attributeStoringPage)
- static void setBundleObject (EAMObject bundleObject)
- static void setBundleSelectItemsGenerated (boolean bundleSelectItemsGenerated)
- static void setCoreObject (EAMObject coreObject)
- static void setCoreSelectItemsGenerated (boolean coreSelectItemsGenerated)

Öffentliche Attribute

- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Private Methoden

- void deleteAssignment (EAMObjectAttributeMatchingStore store)

Private Attribute

- ArrayList< EAMObjectAttributeMatchingStore > attributeMatches
- String bundleAttributeId
- ArrayList< SelectItem > bundleObjectAttributeSelectItems
- String coreAttributeId
- ArrayList< SelectItem > coreObjectAttributeSelectItems
- boolean deleteMappedAndStored = false
- boolean deleteMappedAttribute = false
- boolean deleteMappedAttributeSelected = false
- boolean deleteStoredAttribute = false
- boolean deleteStoredAttributeSelected = false
- DAOFactoryStaticClass factory = DAOFactoryStaticClass.getInstance(userId, password, this.getClass().getName())
- boolean savedEamAttributes = false
- ArrayList< String > selectedBundleAttributes
- String selectedBundleObjectAttribute
- String selectedCoreObjectAttribute

Statische private Attribute

- static boolean attributeMappingPage = true
- static boolean attributeMatchesEmpty = true

- static boolean attributeMatchesGenerated = false
- static boolean attributeStoringPage = false
- static EAMObject bundleObject
- static boolean bundleSelectItemsGenerated = false
- static EAMObject coreObject
- static boolean coreSelectItemsGenerated = false

2.3.2.2 Beschreibung der Konstruktoren und Destruktoren

CoreBundleObjectAttributeMatchingHandler ()

Konstruktor der Bean. Beim Aufruf wird gleich die Liste "attributeMatches" erstellt, um die Datatable mit den bereits bestehenden Verknüpfungen der Objektattribute füllen zu können. Zusätzlich werden alle Flags auf ihren Standard zurückgesetzt.

Definiert in Zeile 68 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt CoreBundleObjectAttributeMatchingHandler.getAttributeMatches() und CoreBundleObjectAttributeMatchingHandler.resetFlags().

2.3.2.3 Dokumentation der Elementfunktionen

void checkDelete ()

Wird ausgeführt, wenn in der Datatable der Löschen-Button in einer Zeile verwendet wird. Dadurch wird eine Sicherheitsabfrage in die View eingebaut, in der das Löschen bestätigt werden muss oder abgebrochen werden kann.

Definiert in Zeile 755 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt CoreBundleObjectAttributeMatchingHandler.attributeMatches, CoreBundleObjectAttributeMatchingHandler.bundleAttributeId, CoreBundleObjectAttributeMatchingHandler.coreAttributeId, CoreBundleObjectAttributeMatchingHandler.deleteMappedAttribute und CoreBundleObjectAttributeMatchingHandler.deleteStoredAttribute.

void checkDeleteMarkedElements ()

Wird ausgeführt, wenn in der View der untere Löschen-Button verwendet wird und eine Checkbox in der Datatable markiert ist. Dadurch wird eine Sicherheitsabfrage in die View eingebaut, in der das Löschen bestätigt werden muss oder abgebrochen werden kann.

Definiert in Zeile 787 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt CoreBundleObjectAttributeMatchingHandler.attributeMatches, CoreBundleObjectAttributeMatchingHandler.bundleAttributeId, CoreBundleObjectAttributeMatchingHandler.coreAttributeId, CoreBundleObjectAttributeMatchingHandler.deleteMappedAndStored,

CoreBundleObjectAttributeMatchingHandler.deleteMappedAttributeSelected und
CoreBundleObjectAttributeMatchingHandler.deleteStoredAttributeSelected.

void deleteAssignment (EAMObjectAttributeMatchingStore store) [private]

Führt das Löschen eines Eintrags in der Verknüpfungstabelle tatsächlich durch. Dabei wird zusätzlich geprüft, ob es sich bei dem EAMObjectAttributeMatchingStore um ein Stored oder Mapped Attribute handelt.

Parameter:

store Der EAMObjectAttributeMatchingStore, dessen Informationen dazu genutzt werden, um eine Verknüpfung oder ein gespeichertes Attribut zu löschen.

Definiert in Zeile 630 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt DAOFactoryStaticClass.attributeDAO,
CoreBundleObjectAttributeMatchingHandler.attributeMatches,
CoreBundleObjectAttributeMatchingHandler.attributeMatchesEmpty,
CoreBundleObjectAttributeMatchingHandler.attributeMatchesGenerated,
CoreBundleObjectAttributeMatchingHandler.bundleSelectItemsGenerated,
CoreBundleObjectAttributeMatchingHandler.coreObject,
CoreBundleObjectAttributeMatchingHandler.coreSelectItemsGenerated,
CoreBundleObjectAttributeMatchingHandler.deleteMappedAndStored,
CoreBundleObjectAttributeMatchingHandler.deleteMappedAttribute,
CoreBundleObjectAttributeMatchingHandler.deleteMappedAttributeSelected,
CoreBundleObjectAttributeMatchingHandler.deleteStoredAttribute,
CoreBundleObjectAttributeMatchingHandler.deleteStoredAttributeSelected,
CoreBundleObjectAttributeMatchingHandler.factory, DAOFactoryStaticClass.objectDAO
und CoreBundleObjectAttributeMatchingHandler.password.

void deleteAssignment ()

Löscht die Verknüpfung von Attribut-Verknüpfungen aus der Datenbank und aus der Liste der Attribut-Mapping-Stores.

Definiert in Zeile 587 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt CoreBundleObjectAttributeMatchingHandler.attributeMatches,
CoreBundleObjectAttributeMatchingHandler.bundleAttributeId und
CoreBundleObjectAttributeMatchingHandler.coreAttributeId.

Wird benutzt von

CoreBundleObjectAttributeMatchingHandler.deleteSelectedAssignments().

void deleteSelectedAssignments ()

Übernimmt das Löschen der in der View selektierten gemappten oder gespeicherten Objekt-Attribute. Dazu werden alle EAMObjectAttributeMatchingStore durchlaufen und geprüft, ob das boolesche Flag "selected" true ist. Nur diese Stores werden gelöscht.

Definiert in Zeile 610 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.attributeMatches` und `CoreBundleObjectAttributeMatchingHandler.deleteAssignment()`.

`ArrayList<EAMObjectAttributeMatchingStore> getAttributeMatches ()`

Ermittelt alle bereits vorhandenen Attribut-Mappings zu den statisch übergebenen EAM-Objekten aus der Datenbank. Im Fehlerfall wird eine leere Liste zurückgegeben.

Rückgabe:

Liste mit allen vorhandenen Attribut-Mappings zu den beiden EAM-Objekten. Im Fehlerfall wird eine leere Liste zurückgegeben.

Definiert in Zeile 502 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.attributeMatches`, `CoreBundleObjectAttributeMatchingHandler.attributeMatchesEmpty`, `CoreBundleObjectAttributeMatchingHandler.attributeMatchesGenerated`, `CoreBundleObjectAttributeMatchingHandler.bundleObject`, `CoreBundleObjectAttributeMatchingHandler.coreObject` und `CoreBundleObjectAttributeMatchingHandler.password`.

Wird benutzt von `CoreBundleObjectAttributeMatchingHandler.CoreBundleObjectAttributeMatchingHandler()`.

`EAMObject getBundleObject ()`

Liefert das aktuelle Bundle-EAMObjekt.

Rückgabe:

Das Bundle-Objekt.

Definiert in Zeile 926 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.bundleObject`.

`ArrayList<SelectedItem> getBundleObjectAttributeSelectItems ()`

Erstellt aus den Attributen des EAM-Objektes des Bundles eine Liste von `SelectItems`, um diese in der `SelectOneListBox` anzeigen zu können. Dabei wird durch ein Boolean-Flag geprüft, ob diese Liste bereits erstellt wurde oder nicht. Wurde sie schon erstellt, wird die Liste nur noch zurückgegeben, ohne diese erneut zu generieren. Tritt ein Fehler in der Generierung auf, wird eine leere Liste zurückgegeben.

Rückgabe:

Die Liste der erstellten `SelectItems`, im Fehlerfall eine leere Liste.

Definiert in Zeile 179 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.bundleObject`,
`CoreBundleObjectAttributeMatchingHandler.bundleObjectAttributeSelectItems`,
`CoreBundleObjectAttributeMatchingHandler.bundleSelectItemsGenerated` und
`CoreBundleObjectAttributeMatchingHandler.password`.

EAMObject getCoreObject ()

Liefert das aktuelle Kern-EAMObjekt.

Rückgabe:

Das Kern-Objekt.

Definiert in Zeile 906 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.coreObject`.

ArrayList<SelectItem> getCoreObjectAttributeSelectItems ()

Erstellt aus den Attributen des EAM-Objektes des Kerns eine Liste von `SelectItems`, um diese in der `SelectOneListBox` anzeigen zu können. Dabei wird durch ein Boolean-Flag geprüft, ob diese Liste bereits erstellt wurde oder nicht. Wurde sie schon erstellt, wird die Liste nur noch zurückgegeben, ohne diese erneut zu generieren. Tritt ein Fehler in der Generierung auf, wird eine leere Liste zurückgegeben.

Rückgabe:

Die Liste der erstellten `SelectItems`, im Fehlerfall eine leere Liste.

Definiert in Zeile 100 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.coreObject`,
`CoreBundleObjectAttributeMatchingHandler.coreObjectAttributeSelectItems`,
`CoreBundleObjectAttributeMatchingHandler.coreSelectItemsGenerated` und
`CoreBundleObjectAttributeMatchingHandler.password`.

ArrayList<String> getSelectedBundleAttributes ()

Definiert in Zeile 1018 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.selectedBundleAttributes`.

String getSelectedBundleObjectAttribute ()

Methode für die Bean, um das ausgewählte Bundleattribut abzufragen.

Rückgabe:

Das ausgewählte Kernattribut.

Definiert in Zeile 886 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.selectedBundleObjectAttribute`.

String getSelectedCoreObjectAttribute ()

Methode für die Bean, um das ausgewählte Kernattribut abzufragen.

Rückgabe:

Das ausgewählte Kernattribut.

Definiert in Zeile 866 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt CoreBundleObjectAttributeMatchingHandler.selectedCoreObjectAttribute.

void goToMappingPage ()

Setzt den Flag für die StoringPage auf False und den Flag für die MappingPage auf True.

Definiert in Zeile 853 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt CoreBundleObjectAttributeMatchingHandler.attributeMappingPage und CoreBundleObjectAttributeMatchingHandler.attributeStoringPage.

void goToStoringPage ()

Setzt den Flag für die StoringPage auf True und den Flag für die MappingPage auf False.

Definiert in Zeile 843 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt CoreBundleObjectAttributeMatchingHandler.attributeMappingPage und CoreBundleObjectAttributeMatchingHandler.attributeStoringPage.

boolean isAttributeMappingPage ()

Definiert in Zeile 998 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt CoreBundleObjectAttributeMatchingHandler.attributeMappingPage.

boolean isAttributeMatchesEmpty ()

Überprüft, ob die Liste der Attribute-Matches leer ist.

Rückgabe:

true, wenn attributeMatches null ist, sonst der jeweilige für attributeMatchesEmpty gesetzte Wert.

Definiert in Zeile 727 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt CoreBundleObjectAttributeMatchingHandler.attributeMatches und CoreBundleObjectAttributeMatchingHandler.attributeMatchesEmpty.

static boolean isAttributeMatchesGenerated () [static]

Definiert in Zeile 962 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt `CoreBundleObjectAttributeMatchingHandler.attributeMatchesGenerated`.

boolean isAttributeStoringPage ()

Definiert in Zeile 1008 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.attributeStoringPage`.

static boolean isBundleSelectItemsGenerated () [static]

Definiert in Zeile 941 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.bundleSelectItemsGenerated`.

static boolean isCoreSelectItemsGenerated () [static]

Definiert in Zeile 952 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.coreSelectItemsGenerated`.

boolean isDeleteMappedAndStored ()

Definiert in Zeile 1029 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.deleteMappedAndStored`.

boolean isDeleteMappedAttribute ()

Definiert in Zeile 988 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.deleteMappedAttribute`.

boolean isDeleteMappedAttributeSelected ()

Definiert in Zeile 1050 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.deleteMappedAttributeSelected`.

boolean isDeleteStoredAttribute ()

Definiert in Zeile 978 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.deleteStoredAttribute`.

boolean isDeleteStoredAttributeSelected ()

Definiert in Zeile 1039 der Datei `CoreBundleObjectAttributeMatchingHandler.java`.

Benutzt `CoreBundleObjectAttributeMatchingHandler.deleteStoredAttributeSelected`.

boolean isSizeGreaterTen ()

Ermittelt die Größe der Liste von Attribut-Mappings und gibt zurück, ob die Liste > 10 ist.

Rückgabe:

true, wenn die Liste > 10, false sonst.

Definiert in Zeile 741 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt CoreBundleObjectAttributeMatchingHandler.attributeMatches.

void noDelete ()

Setzt die Boolean-Flags, die für das Löschen von Verknüpfungen verwendet werden, wieder zurück, so dass keine Nachfragen für das Löschen mehr vorhanden sind. Dies bricht den Vorgang des Löschens ab.

Definiert in Zeile 829 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt CoreBundleObjectAttributeMatchingHandler.deleteMappedAndStored, CoreBundleObjectAttributeMatchingHandler.deleteMappedAttribute, CoreBundleObjectAttributeMatchingHandler.deleteMappedAttributeSelected, CoreBundleObjectAttributeMatchingHandler.deleteStoredAttribute und CoreBundleObjectAttributeMatchingHandler.deleteStoredAttributeSelected.

void resetFlags ()

Setzt alle Boolean-Flags auf ihren Standard-Wert zurück.

Definiert in Zeile 77 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt CoreBundleObjectAttributeMatchingHandler.attributeMatchesEmpty, CoreBundleObjectAttributeMatchingHandler.attributeMatchesGenerated, CoreBundleObjectAttributeMatchingHandler.bundleSelectItemsGenerated, CoreBundleObjectAttributeMatchingHandler.coreSelectItemsGenerated, CoreBundleObjectAttributeMatchingHandler.deleteMappedAndStored, CoreBundleObjectAttributeMatchingHandler.deleteMappedAttribute, CoreBundleObjectAttributeMatchingHandler.deleteMappedAttributeSelected, CoreBundleObjectAttributeMatchingHandler.deleteStoredAttribute und CoreBundleObjectAttributeMatchingHandler.deleteStoredAttributeSelected.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.CoreBundleObjectAttributeMatchingHandler().

void saveAttributeAssignment ()

Speichert die Attributzuweisungen. Dazu muss ein Kern-Attribut und ein Bundle-Attribut in der jeweiligen ListBox ausgewählt werden. Diese Verknüpfung wird in der DB gespeichert und die entsprechenden Einträge aus den Listboxen entfernt. Gleichzeitig wird

ein Eintrag in der Liste der Objekt-Attribute-Verknüpfungen hinzugefügt, um diese in der Datatable der View anzuzeigen.

Definiert in Zeile 255 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt CoreBundleObjectAttributeMatchingHandler.attributeMatches,
CoreBundleObjectAttributeMatchingHandler.attributeMatchesEmpty,
CoreBundleObjectAttributeMatchingHandler.bundleAttributeId,
CoreBundleObjectAttributeMatchingHandler.bundleObject,
CoreBundleObjectAttributeMatchingHandler.bundleObjectAttributeSelectItems,
CoreBundleObjectAttributeMatchingHandler.coreAttributeId,
CoreBundleObjectAttributeMatchingHandler.coreObject,
CoreBundleObjectAttributeMatchingHandler.coreObjectAttributeSelectItems,
CoreBundleObjectAttributeMatchingHandler.password,
CoreBundleObjectAttributeMatchingHandler.selectedBundleObjectAttribute und
CoreBundleObjectAttributeMatchingHandler.selectedCoreObjectAttribute.

static void setAttributeMappingPage (boolean attributeMappingPage) [static]

Definiert in Zeile 1003 der Datei CoreBundleObjectAttributeMatchingHandler.java.

static void setAttributeMatchesEmpty (boolean attributeMatchesEmpty) [static]

Definiert in Zeile 973 der Datei CoreBundleObjectAttributeMatchingHandler.java.

static void setAttributeMatchesGenerated (boolean attributeMatchesGenerated) [static]

Definiert in Zeile 967 der Datei CoreBundleObjectAttributeMatchingHandler.java.

static void setAttributeStoringPage (boolean attributeStoringPage) [static]

Definiert in Zeile 1013 der Datei CoreBundleObjectAttributeMatchingHandler.java.

static void setBundleObject (EAMObject bundleObject) [static]

Setzt das Bundle-EAMObjekt.

Parameter:

bundleObject Das zu setzende Bundle-EAMObjekt.

Definiert in Zeile 936 der Datei CoreBundleObjectAttributeMatchingHandler.java.

static void setBundleSelectItemsGenerated (boolean bundleSelectItemsGenerated) [static]

Definiert in Zeile 946 der Datei CoreBundleObjectAttributeMatchingHandler.java.

static void setCoreObject (EAMObject coreObject) [static]

Setzt das Kern-EAMObjekt.

Parameter:

coreObject Das zu setzende Kern-EAMObjekt.

Definiert in Zeile 916 der Datei CoreBundleObjectAttributeMatchingHandler.java.

static void setCoreSelectItemsGenerated (boolean coreSelectItemsGenerated) [static]

Definiert in Zeile 957 der Datei CoreBundleObjectAttributeMatchingHandler.java.

void setDeleteMappedAndStored (boolean deleteMappedAndStored)

Definiert in Zeile 1034 der Datei CoreBundleObjectAttributeMatchingHandler.java.

void setDeleteMappedAttribute (boolean deleteMappedAttribute)

Definiert in Zeile 993 der Datei CoreBundleObjectAttributeMatchingHandler.java.

void setDeleteMappedAttributeSelected (boolean deleteMappedAttributeSelected)

Definiert in Zeile 1055 der Datei CoreBundleObjectAttributeMatchingHandler.java.

void setDeleteStoredAttribute (boolean deleteStoredAttribute)

Definiert in Zeile 983 der Datei CoreBundleObjectAttributeMatchingHandler.java.

void setDeleteStoredAttributeSelected (boolean deleteStoredAttributeSelected)

Definiert in Zeile 1044 der Datei CoreBundleObjectAttributeMatchingHandler.java.

void setSelectedBundleAttributes (ArrayList< String > selectedBundleAttributes)

Definiert in Zeile 1023 der Datei CoreBundleObjectAttributeMatchingHandler.java.

void setSelectedBundleObjectAttribute (String selectedBundleObjectAttribute)

Methode für die Bean, um den ausgewählten Eintrag der Kernattribute speichern und für die Weiterverarbeitung bereitstellen zu können.

Definiert in Zeile 896 der Datei CoreBundleObjectAttributeMatchingHandler.java.

void setSelectedCoreObjectAttribute (String selectedCoreObjectAttribute)

Methode für die Bean, um den ausgewählten Eintrag der Kernattribute speichern und für die Weiterverarbeitung bereitstellen zu können.

Definiert in Zeile 876 der Datei CoreBundleObjectAttributeMatchingHandler.java.

void storeAttributesToObject ()

Fügt dem EAM-Objekt des Kerns das in der Liste ausgewählte Attribut hinzu und speichert diese Zuweisung in der DB. Das ausgewählte Attribut wird aus der Listbox entfernt und ein Eintrag in der Liste der Attribut-Verknüpfungen hinzugefügt.

Definiert in Zeile 341 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Benutzt CoreBundleObjectAttributeMatchingHandler.attributeMatches,
CoreBundleObjectAttributeMatchingHandler.attributeMatchesEmpty,
DAOFactoryStaticClass.attributeTypeDAO,
CoreBundleObjectAttributeMatchingHandler.bundleAttributeId,
CoreBundleObjectAttributeMatchingHandler.bundleObject,
CoreBundleObjectAttributeMatchingHandler.bundleObjectAttributeSelectItems,
CoreBundleObjectAttributeMatchingHandler.coreObject,
CoreBundleObjectAttributeMatchingHandler.factory, DAOFactoryStaticClass.objectDAO,
CoreBundleObjectAttributeMatchingHandler.password und
CoreBundleObjectAttributeMatchingHandler.selectedBundleAttributes.

2.3.2.4 Dokumentation der Datenelemente**boolean attributeMappingPage = true [static, private]**

Definiert in Zeile 51 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.goToMappingPage(),
CoreBundleObjectAttributeMatchingHandler.goToStoringPage() und
CoreBundleObjectAttributeMatchingHandler.isAttributeMappingPage().

ArrayList<EAMObjectAttributeMatchingStore> attributeMatches [private]

Definiert in Zeile 42 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.checkDelete(),
CoreBundleObjectAttributeMatchingHandler.checkDeleteMarkedElements(),
CoreBundleObjectAttributeMatchingHandler.deleteAssignment(),
CoreBundleObjectAttributeMatchingHandler.deleteSelectedAssignments(),
CoreBundleObjectAttributeMatchingHandler.getAttributeMatches(),
CoreBundleObjectAttributeMatchingHandler.isAttributeMatchesEmpty(),
CoreBundleObjectAttributeMatchingHandler.isSizeGreaterTen(),
CoreBundleObjectAttributeMatchingHandler.saveAttributeAssignment() und
CoreBundleObjectAttributeMatchingHandler.storeAttributesToObject().

boolean attributeMatchesEmpty = true [static, private]

Definiert in Zeile 50 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.deleteAssignment(), CoreBundleObjectAttributeMatchingHandler.getAttributeMatches(), CoreBundleObjectAttributeMatchingHandler.isAttributeMatchesEmpty(), CoreBundleObjectAttributeMatchingHandler.resetFlags(), CoreBundleObjectAttributeMatchingHandler.saveAttributeAssignment() und CoreBundleObjectAttributeMatchingHandler.storeAttributesToObject().

boolean attributeMatchesGenerated = false [static, private]

Definiert in Zeile 49 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.deleteAssignment(), CoreBundleObjectAttributeMatchingHandler.getAttributeMatches(), CoreBundleObjectAttributeMatchingHandler.isAttributeMatchesGenerated() und CoreBundleObjectAttributeMatchingHandler.resetFlags().

boolean attributeStoringPage = false [static, private]

Definiert in Zeile 52 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.goToMappingPage(), CoreBundleObjectAttributeMatchingHandler.goToStoringPage() und CoreBundleObjectAttributeMatchingHandler.isAttributeStoringPage().

String bundleAttributeId [private]

Definiert in Zeile 60 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.checkDelete(), CoreBundleObjectAttributeMatchingHandler.checkDeleteMarkedElements(), CoreBundleObjectAttributeMatchingHandler.deleteAssignment(), CoreBundleObjectAttributeMatchingHandler.saveAttributeAssignment() und CoreBundleObjectAttributeMatchingHandler.storeAttributesToObject().

EAMObject bundleObject [static, private]

Definiert in Zeile 44 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.getAttributeMatches(), CoreBundleObjectAttributeMatchingHandler.getBundleObject(), CoreBundleObjectAttributeMatchingHandler.getBundleObjectAttributeSelectItems(), CoreBundleObjectAttributeMatchingHandler.saveAttributeAssignment() und CoreBundleObjectAttributeMatchingHandler.storeAttributesToObject().

ArrayList<SelectedItem> bundleObjectAttributeSelectItems [private]

Definiert in Zeile 39 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von

CoreBundleObjectAttributeMatchingHandler.getBundleObjectAttributeSelectItems(),
CoreBundleObjectAttributeMatchingHandler.saveAttributeAssignment() und
CoreBundleObjectAttributeMatchingHandler.storeAttributesToObject().

boolean bundleSelectItemsGenerated = false [static, private]

Definiert in Zeile 47 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.deleteAssignment(),
CoreBundleObjectAttributeMatchingHandler.getBundleObjectAttributeSelectItems(),
CoreBundleObjectAttributeMatchingHandler.isBundleSelectItemsGenerated() und
CoreBundleObjectAttributeMatchingHandler.resetFlags().

String coreAttributeId [private]

Definiert in Zeile 61 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.checkDelete(),
CoreBundleObjectAttributeMatchingHandler.checkDeleteMarkedElements(),
CoreBundleObjectAttributeMatchingHandler.deleteAssignment() und
CoreBundleObjectAttributeMatchingHandler.saveAttributeAssignment().

EAMObject coreObject [static, private]

Definiert in Zeile 45 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.deleteAssignment(),
CoreBundleObjectAttributeMatchingHandler.getAttributeMatches(),
CoreBundleObjectAttributeMatchingHandler.getCoreObject(),
CoreBundleObjectAttributeMatchingHandler.getCoreObjectAttributeSelectItems(),
CoreBundleObjectAttributeMatchingHandler.saveAttributeAssignment() und
CoreBundleObjectAttributeMatchingHandler.storeAttributesToObject().

ArrayList<SelectItem> coreObjectAttributeSelectItems [private]

Definiert in Zeile 35 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von

CoreBundleObjectAttributeMatchingHandler.getCoreObjectAttributeSelectItems() und
CoreBundleObjectAttributeMatchingHandler.saveAttributeAssignment().

boolean coreSelectItemsGenerated = false [static, private]

Definiert in Zeile 48 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.deleteAssignment(),
CoreBundleObjectAttributeMatchingHandler.getCoreObjectAttributeSelectItems(),

CoreBundleObjectAttributeMatchingHandler.isCoreSelectItemsGenerated() und
CoreBundleObjectAttributeMatchingHandler.resetFlags().

boolean deleteMappedAndStored = false [private]

Definiert in Zeile 58 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von

CoreBundleObjectAttributeMatchingHandler.checkDeleteMarkedElements(),
CoreBundleObjectAttributeMatchingHandler.deleteAssignment(),
CoreBundleObjectAttributeMatchingHandler.isDeleteMappedAndStored(),
CoreBundleObjectAttributeMatchingHandler.noDelete() und
CoreBundleObjectAttributeMatchingHandler.resetFlags().

boolean deleteMappedAttribute = false [private]

Definiert in Zeile 55 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.checkDelete(),
CoreBundleObjectAttributeMatchingHandler.deleteAssignment(),
CoreBundleObjectAttributeMatchingHandler.isDeleteMappedAttribute(),
CoreBundleObjectAttributeMatchingHandler.noDelete() und
CoreBundleObjectAttributeMatchingHandler.resetFlags().

boolean deleteMappedAttributeSelected = false [private]

Definiert in Zeile 57 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von

CoreBundleObjectAttributeMatchingHandler.checkDeleteMarkedElements(),
CoreBundleObjectAttributeMatchingHandler.deleteAssignment(),
CoreBundleObjectAttributeMatchingHandler.isDeleteMappedAttributeSelected(),
CoreBundleObjectAttributeMatchingHandler.noDelete() und
CoreBundleObjectAttributeMatchingHandler.resetFlags().

boolean deleteStoredAttribute = false [private]

Definiert in Zeile 54 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.checkDelete(),
CoreBundleObjectAttributeMatchingHandler.deleteAssignment(),
CoreBundleObjectAttributeMatchingHandler.isDeleteStoredAttribute(),
CoreBundleObjectAttributeMatchingHandler.noDelete() und
CoreBundleObjectAttributeMatchingHandler.resetFlags().

boolean deleteStoredAttributeSelected = false [private]

Definiert in Zeile 56 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von

CoreBundleObjectAttributeMatchingHandler.checkDeleteMarkedElements(),
CoreBundleObjectAttributeMatchingHandler.deleteAssignment(),
CoreBundleObjectAttributeMatchingHandler.isDeleteStoredAttributeSelected(),
CoreBundleObjectAttributeMatchingHandler.noDelete() und
CoreBundleObjectAttributeMatchingHandler.resetFlags().

**DAOFactoryStaticClass factory = DAOFactoryStaticClass.getInstance(userId,
password, this.getClass().getName()) [private]**

Definiert in Zeile 33 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.deleteAssignment() und
CoreBundleObjectAttributeMatchingHandler.storeAttributesToObject().

**String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")**

Definiert in Zeile 31 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von CoreBundleObjectAttributeMatchingHandler.deleteAssignment(),
CoreBundleObjectAttributeMatchingHandler.getAttributeMatches(),
CoreBundleObjectAttributeMatchingHandler.getBundleObjectAttributeSelectItems(),
CoreBundleObjectAttributeMatchingHandler.getCoreObjectAttributeSelectItems(),
CoreBundleObjectAttributeMatchingHandler.saveAttributeAssignment() und
CoreBundleObjectAttributeMatchingHandler.storeAttributesToObject().

boolean savedEamAttributes = false [private]

Definiert in Zeile 53 der Datei CoreBundleObjectAttributeMatchingHandler.java.

ArrayList<String> selectedBundleAttributes [private]

Definiert in Zeile 37 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von

CoreBundleObjectAttributeMatchingHandler.getSelectedBundleAttributes() und
CoreBundleObjectAttributeMatchingHandler.storeAttributesToObject().

String selectedBundleObjectAttribute [private]

Definiert in Zeile 40 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von

CoreBundleObjectAttributeMatchingHandler.getSelectedBundleObjectAttribute() und
CoreBundleObjectAttributeMatchingHandler.saveAttributeAssignment().

String selectedCoreObjectAttribute [private]

Definiert in Zeile 36 der Datei CoreBundleObjectAttributeMatchingHandler.java.

Wird benutzt von

CoreBundleObjectAttributeMatchingHandler.getSelectedCoreObjectAttribute() und
CoreBundleObjectAttributeMatchingHandler.saveAttributeAssignment().

String userId =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Definiert in Zeile 30 der Datei CoreBundleObjectAttributeMatchingHandler.java.

2.3.3 CoreBundleObjectMatchingHandler Klassenreferenz

2.3.3.1 Ausführliche Beschreibung

ManagedBean für das Mapping der EAM-Objekte des Kerns und des ausgewählten Bundles sowie und die Speicherung der Bundle-Objekte in den Kern. Enthält gesamte Logik, die für diese Aufgaben benötigt werden.

Autor:

Jens Henkel

Definiert in Zeile 29 der Datei CoreBundleObjectMatchingHandler.java.

Öffentliche Methoden

- CoreBundleObjectMatchingHandler ()
- ArrayList< SelectItem > getCoreObjectSelectItems ()
- void resetFlags ()

Öffentliche Attribute

- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Private Attribute

- ArrayList< EAMCategory > allCategories
- String bundleObjectId

- `ArrayList< SelectItem > bundleObjectSelectItems`
- `boolean bundleSelectItemsEmpty = false`
- `String coreObjectId`
- `ArrayList< EAMObject > coreObjects`
- `ArrayList< SelectItem > coreObjectSelectItems`
- `boolean deleteMappedAndStored = false`
- `boolean deleteMappedObject = false`
- `boolean deleteMappedObjectSelected = false`
- `boolean deleteStoredObject = false`
- `boolean deleteStoredObjectSelected = false`
- `DAOFactoryStaticClass factory = DAOFactoryStaticClass.getInstance(userId, password, this.getClass().getName())`
- `ArrayList< SelectItem > metamodelSelectItems`
- `ArrayList< EAMObjectMatchingStore > objectMatches`
- `boolean objectMatchesEmpty = true`
- `boolean objectMatchesWorkaround = false`
- `boolean savedEamObjects = false`
- `String selectedBundleObject`
- `ArrayList< String > selectedBundleObjects = new ArrayList<String>()`
- `String selectedCoreObject`
- `ArrayList< String > selectedMetamodels = new ArrayList<String>()`

Statische private Attribute

- `static LinkedList< EAMObject > bundleObjects`
- `static boolean bundleSelectItemsGenerated = false`
- `static boolean coreSelectItemsGenerated = false`
- `static boolean metamodelSelectItemsGenerated = false`
- `static boolean objectMappingPage = true`
- `static boolean objectMatchesGenerated = false`
- `static boolean objectStoringPage = false`

2.3.3.2 Beschreibung der Konstruktoren und Destruktoren

CoreBundleObjectMatchingHandler ()

Konstruktor der Bean. Beim Erzeugen der Bean wird die Liste bereits vorhandener Objekt-Verknüpfungen geprüft und erstellt. Zusätzlich werden alle Boolean-Flags auf ihren Standard-Wert zurückgesetzt.

Definiert in Zeile 78 der Datei `CoreBundleObjectMatchingHandler.java`.

Benutzt CoreBundleObjectMatchingHandler.resetFlags().

2.3.3.3 Dokumentation der Elementfunktionen

ArrayList<SelectItem> getCoreObjectSelectItems ()

Erstellt für alle EAMObjekte des Kerns eine Liste von SelectItems. Prüft, ob die Liste noch null ist und erzeugt gegebenenfalls ein neues Objekt. Wurde die Liste bereits erstellt, wird diese Liste nur zurückgegeben und nicht neu generiert.

Rückgabe:

Die Liste der SelectItems, eine leere Liste im Fehlerfall.

Definiert in Zeile 111 der Datei CoreBundleObjectMatchingHandler.java.

Benutzt CoreBundleObjectMatchingHandler.coreObjects,
CoreBundleObjectMatchingHandler.coreObjectSelectItems,
CoreBundleObjectMatchingHandler.coreSelectItemsGenerated,
CoreBundleObjectMatchingHandler.factory, DAOFactoryStaticClass.objectDAO und
CoreBundleObjectMatchingHandler.password.

void resetFlags ()

Setzt alle Boolean-Flags auf ihren Standard-Wert zurück.

Definiert in Zeile 87 der Datei CoreBundleObjectMatchingHandler.java.

Benutzt CoreBundleObjectMatchingHandler.bundleSelectItemsEmpty,
CoreBundleObjectMatchingHandler.bundleSelectItemsGenerated,
CoreBundleObjectMatchingHandler.coreSelectItemsGenerated,
CoreBundleObjectMatchingHandler.deleteMappedAndStored,
CoreBundleObjectMatchingHandler.deleteMappedObject,
CoreBundleObjectMatchingHandler.deleteMappedObjectSelected,
CoreBundleObjectMatchingHandler.deleteStoredObject,
CoreBundleObjectMatchingHandler.deleteStoredObjectSelected,
CoreBundleObjectMatchingHandler.metamodelSelectItemsGenerated,
CoreBundleObjectMatchingHandler.objectMatchesEmpty und
CoreBundleObjectMatchingHandler.objectMatchesGenerated.

Wird benutzt von

CoreBundleObjectMatchingHandler.CoreBundleObjectMatchingHandler().

2.3.3.4 Dokumentation der Datenelemente

ArrayList<EAMCategory> allCategories [private]

Definiert in Zeile 47 der Datei CoreBundleObjectMatchingHandler.java.

String bundleObjectId [private]

Definiert in Zeile 51 der Datei CoreBundleObjectMatchingHandler.java.

LinkedList<EAMObject> bundleObjects [static, private]

Definiert in Zeile 41 der Datei CoreBundleObjectMatchingHandler.java.

ArrayList<SelectItem> bundleObjectSelectItems [private]

Definiert in Zeile 40 der Datei CoreBundleObjectMatchingHandler.java.

boolean bundleSelectItemsEmpty = false [private]

Definiert in Zeile 57 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.resetFlags().

boolean bundleSelectItemsGenerated = false [static, private]

Definiert in Zeile 54 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.resetFlags().

String coreObjectId [private]

Definiert in Zeile 52 der Datei CoreBundleObjectMatchingHandler.java.

ArrayList<EAMObject> coreObjects [private]

Definiert in Zeile 37 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.getCoreObjectSelectItems().

ArrayList<SelectItem> coreObjectSelectItems [private]

Definiert in Zeile 36 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.getCoreObjectSelectItems().

boolean coreSelectItemsGenerated = false [static, private]

Definiert in Zeile 55 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.getCoreObjectSelectItems() und CoreBundleObjectMatchingHandler.resetFlags().

boolean deleteMappedAndStored = false [private]

Definiert in Zeile 66 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.resetFlags().

boolean deleteMappedObject = false [private]

Definiert in Zeile 68 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.resetFlags().

boolean deleteMappedObjectSelected = false [private]

Definiert in Zeile 70 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.resetFlags().

boolean deleteStoredObject = false [private]

Definiert in Zeile 67 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.resetFlags().

boolean deleteStoredObjectSelected = false [private]

Definiert in Zeile 69 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.resetFlags().

DAOFactoryStaticClass factory = DAOFactoryStaticClass.getInstance(userId, password, this.getClass().getName()) [private]

Definiert in Zeile 34 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.getCoreObjectSelectItems().

ArrayList<SelectedItem> metamodelSelectItems [private]

Definiert in Zeile 46 der Datei CoreBundleObjectMatchingHandler.java.

boolean metamodelSelectItemsGenerated = false [static, private]

Definiert in Zeile 62 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.resetFlags().

boolean objectMappingPage = true [static, private]

Definiert in Zeile 60 der Datei CoreBundleObjectMatchingHandler.java.

ArrayList<EAMObjectMatchingStore> objectMatches [private]

Definiert in Zeile 44 der Datei CoreBundleObjectMatchingHandler.java.

boolean objectMatchesEmpty = true [private]

Definiert in Zeile 58 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.resetFlags().

boolean objectMatchesGenerated = false [static, private]

Definiert in Zeile 56 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.resetFlags().

boolean objectMatchesWorkaround = false [private]

Definiert in Zeile 64 der Datei CoreBundleObjectMatchingHandler.java.

boolean objectStoringPage = false [static, private]

Definiert in Zeile 61 der Datei CoreBundleObjectMatchingHandler.java.

String password =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")

Definiert in Zeile 32 der Datei CoreBundleObjectMatchingHandler.java.

Wird benutzt von CoreBundleObjectMatchingHandler.getCoreObjectSelectItems().

boolean savedEamObjects = false [private]

Definiert in Zeile 63 der Datei CoreBundleObjectMatchingHandler.java.

String selectedBundleObject [private]

Definiert in Zeile 42 der Datei CoreBundleObjectMatchingHandler.java.

ArrayList<String> selectedBundleObjects = new ArrayList<String>() [private]

Definiert in Zeile 49 der Datei CoreBundleObjectMatchingHandler.java.

String selectedCoreObject [private]

Definiert in Zeile 38 der Datei CoreBundleObjectMatchingHandler.java.

```
ArrayList<String> selectedMetamodels = new ArrayList<String>() [private]
```

Definiert in Zeile 48 der Datei CoreBundleObjectMatchingHandler.java.

```
String userId =  
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")
```

Definiert in Zeile 31 der Datei CoreBundleObjectMatchingHandler.java.

2.3.4 CoreBundleRelationAttributeMatchingHandler Klassenreferenz

2.3.4.1 Ausführliche Beschreibung

ManagedBean für das Mapping der Relationen-Attribute einer Kern- und einer Bundle-Relation sowie das Anfügen von Attributen an eine Kernrelation.

Autor:

Jens Henkel

Definiert in Zeile 27 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Öffentliche Methoden

- String checkDelete ()
- void checkDeleteMarkedElements ()
- CoreBundleRelationAttributeMatchingHandler ()
- void deleteAssignment ()
- void deleteSelectedAssignments ()
- ArrayList< EAMRelationAttributeMatchingStore > getAttributeMatches ()
- EAMRelation getBundleRelation ()
- ArrayList< SelectItem > getBundleRelationAttributeSelectItems ()
- EAMRelation getCoreRelation ()
- ArrayList< SelectItem > getCoreRelationAttributeSelectItems ()
- String getSelectedBundleRelationAttribute ()
- ArrayList< String > getSelectedBundleRelationAttributes ()
- String getSelectedCoreRelationAttribute ()
- void goToMappingPage ()
- void goToStoringPage ()
- boolean isAttributeMappingPage ()
- boolean isAttributeMatchesEmpty ()
- boolean isAttributeStoringPage ()

- boolean isDeleteMappedAndStored ()
- boolean isDeleteMappedAttribute ()
- boolean isDeleteMappedAttributeSelected ()
- boolean isDeleteStoredAttribute ()
- boolean isDeleteStoredAttributeSelected ()
- boolean isSizeGreaterTen ()
- void noDelete ()
- void resetFlags ()
- void saveAttributeAssignment ()
- void setDeleteMappedAndStored (boolean deleteMappedAndStored)
- void setDeleteMappedAttribute (boolean deleteMappedAttribute)
- void setDeleteMappedAttributeSelected (boolean deleteMappedAttributeSelected)
- void setDeleteStoredAttribute (boolean deleteStoredAttribute)
- void setDeleteStoredAttributeSelected (boolean deleteStoredAttributeSelected)
- void setSelectedBundleRelationAttribute (String selectedBundleRelationAttribute)
- void setSelectedBundleRelationAttributes (ArrayList< String > selectedBundleRelationAttributes)
- void setSelectedCoreRelationAttribute (String selectedCoreRelationAttribute)
- void storeAttributesToRelation ()

Öffentliche, statische Methoden

- static boolean isAttributeMatchesGenerated ()
- static boolean isBundleSelectItemsGenerated ()
- static boolean isCoreSelectItemsGenerated ()
- static void setAttributeMappingPage (boolean attributeMappingPage)
- static void setAttributeMatchesEmpty (boolean attributeMatchesEmpty)
- static void setAttributeMatchesGenerated (boolean attributeMatchesGenerated)
- static void setAttributeStoringPage (boolean attributeStoringPage)
- static void setBundleRelation (EAMRelation bundleRelation)
- static void setBundleSelectItemsGenerated (boolean bundleSelectItemsGenerated)
- static void setCoreRelation (EAMRelation coreRelation)
- static void setCoreSelectItemsGenerated (boolean coreSelectItemsGenerated)

Öffentliche Attribute

- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String userId =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Private Methoden

- void deleteAssignment (EAMRelationAttributeMatchingStore store)

Private Attribute

- ArrayList< EAMRelationAttributeMatchingStore > attributeMatches
- String bundleAttributeId
- ArrayList< SelectItem > bundleRelationAttributeSelectItems
- String coreAttributeId
- ArrayList< SelectItem > coreRelationAttributeSelectItems
- boolean deleteMappedAndStored = false
- boolean deleteMappedAttribute = false
- boolean deleteMappedAttributeSelected = false
- boolean deleteStoredAttribute = false
- boolean deleteStoredAttributeSelected = false
- DAOFactoryStaticClass factory = DAOFactoryStaticClass.getInstance(userId, password, this.getClass().getName())
- String selectedBundleRelationAttribute
- ArrayList< String > selectedBundleRelationAttributes
- String selectedCoreRelationAttribute

Statische private Attribute

- static boolean attributeMappingPage = true
- static boolean attributeMatchesEmpty = true
- static boolean attributeMatchesGenerated = false
- static boolean attributeStoringPage = false
- static EAMRelation bundleRelation
- static boolean bundleSelectItemsGenerated = false
- static EAMRelation coreRelation
- static boolean coreSelectItemsGenerated = false

2.3.4.2 Beschreibung der Konstruktoren und Destruktoren

CoreBundleRelationAttributeMatchingHandler ()

Konstruktor der Bean. Beim Erzeugen der Bean wird die Liste bereits vorhandener

Relations-Attribut-Verknüpfungen geprüft und erstellt. Zusätzlich werden alle Boolean-Flags auf ihren Standard-Wert zurückgesetzt.

Definiert in Zeile 68 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Benutzt `CoreBundleRelationAttributeMatchingHandler.getAttributeMatches()` und `CoreBundleRelationAttributeMatchingHandler.resetFlags()`.

2.3.4.3 Dokumentation der Elementfunktionen

String checkDelete ()

Wird ausgeführt, nachdem der Löschen-Button innerhalb der Datatbale betätigt wird. Abhängig davon, ob es sich um ein `StoredRelation` handelt oder nicht, wird ein Boolean-Flag gesetzt, der eine Sicherheitsabfrage in der View anzeigt.

Definiert in Zeile 732 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Benutzt `CoreBundleRelationAttributeMatchingHandler.attributeMatches`, `CoreBundleRelationAttributeMatchingHandler.bundleAttributeId`, `CoreBundleRelationAttributeMatchingHandler.coreAttributeId`, `CoreBundleRelationAttributeMatchingHandler.deleteMappedAttribute` und `CoreBundleRelationAttributeMatchingHandler.deleteStoredAttribute`.

void checkDeleteMarkedElements ()

Wird ausgeführt, nachdem der Löschen-Button unterhalb der Datatable betätigt wird. Dadurch werden Boolean-Flags gesetzt, die abhängig von den markierten Checkboxes sind. Dadurch wird eine Sicherheitsabfrage in der View dargestellt.

Definiert in Zeile 766 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Benutzt `CoreBundleRelationAttributeMatchingHandler.attributeMatches`, `CoreBundleRelationAttributeMatchingHandler.bundleAttributeId`, `CoreBundleRelationAttributeMatchingHandler.coreAttributeId`, `CoreBundleRelationAttributeMatchingHandler.deleteMappedAndStored`, `CoreBundleRelationAttributeMatchingHandler.deleteMappedAttributeSelected` und `CoreBundleRelationAttributeMatchingHandler.deleteStoredAttributeSelected`.

void deleteAssignment (EAMRelationAttributeMatchingStore store) [private]

Führt das Löschen eines Eintrags in der Verknüpfungstabelle tatsächlich durch. Dabei wird zusätzlich geprüft, ob es sich bei dem `EAMRelationAttributeMatchingStore` um ein `Stored` oder `Mapped Attribute` handelt.

Parameter:

store Der `EAMRelationAttributeMatchingStore`, dessen Informationen dazu genutzt werden, um eine Verknüpfung oder ein gespeichertes Attribut zu löschen.

Definiert in Zeile 223 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Benutzt DAOFactoryStaticClass.attributeDAO,
CoreBundleRelationAttributeMatchingHandler.attributeMatches,
CoreBundleRelationAttributeMatchingHandler.attributeMatchesEmpty,
CoreBundleRelationAttributeMatchingHandler.attributeMatchesGenerated,
CoreBundleRelationAttributeMatchingHandler.bundleSelectItemsGenerated,
CoreBundleRelationAttributeMatchingHandler.coreRelation,
CoreBundleRelationAttributeMatchingHandler.coreSelectItemsGenerated,
CoreBundleRelationAttributeMatchingHandler.deleteMappedAndStored,
CoreBundleRelationAttributeMatchingHandler.deleteMappedAttribute,
CoreBundleRelationAttributeMatchingHandler.deleteMappedAttributeSelected,
CoreBundleRelationAttributeMatchingHandler.deleteStoredAttribute,
CoreBundleRelationAttributeMatchingHandler.deleteStoredAttributeSelected,
CoreBundleRelationAttributeMatchingHandler.factory,
CoreBundleRelationAttributeMatchingHandler.password und
DAOFactoryStaticClass.relationDAO.

void deleteAssignment ()

Löscht einen einzelnen AttributMatchingStore aus der Liste der AttributeMatches.

Definiert in Zeile 182 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.attributeMatches,
CoreBundleRelationAttributeMatchingHandler.bundleAttributeId und
CoreBundleRelationAttributeMatchingHandler.coreAttributeId.

Wird benutzt von

CoreBundleRelationAttributeMatchingHandler.deleteSelectedAssignments().

void deleteSelectedAssignments ()

Übernimmt das Löschen der in der View selektierten gemappten oder gespeicherten Relations-Attribute. Dazu werden alle EAMRelationAttributeMatchingStore durchlaufen und geprüft, ob das boolesche Flag "selected" true ist. Nur diese Stores werden gelöscht.

Definiert in Zeile 203 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.attributeMatches und
CoreBundleRelationAttributeMatchingHandler.deleteAssignment().

ArrayList<EAMRelationAttributeMatchingStore> getAttributeMatches ()

Ermittelt eine Liste von bereits erstellten Attribut-Verknüpfungen der ausgewählten Relation.

Rückgabe:

Eine Liste der bereits verknüpften Relationsattribute oder eine leere Liste im Fehlerfall.

Definiert in Zeile 95 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt `CoreBundleRelationAttributeMatchingHandler.attributeMatches`,
`CoreBundleRelationAttributeMatchingHandler.attributeMatchesEmpty`,
`CoreBundleRelationAttributeMatchingHandler.attributeMatchesGenerated`,
`CoreBundleRelationAttributeMatchingHandler.bundleRelation`,
`CoreBundleRelationAttributeMatchingHandler.coreRelation` und
`CoreBundleRelationAttributeMatchingHandler.password`.

Wird benutzt von `CoreBundleRelationAttributeMatchingHandler.CoreBundleRelationAttributeMatchingHandler()`.

EAMRelation getBundleRelation ()

Definiert in Zeile 859 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Benutzt `CoreBundleRelationAttributeMatchingHandler.bundleRelation`.

ArrayList<SelectItem> getBundleRelationAttributeSelectItems ()

Generiert die `SelectItems` der Relations-Attribute einer Modul-Relation für die `SelectListbox` der View. Dabei wird zusätzlich geprüft, ob zu einem Attribut bereits ein Eintrag in der Verknüpfungs-Tabelle der Relations-Attribute vorhanden ist. Ist dies der Fall, wird für dieses Attribut kein `SelectItem` erstellt.

Rückgabe:

Die vollständig generierte Liste der `SelectItems`, eine leere Liste im Fehlerfall.

Definiert in Zeile 586 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Benutzt `CoreBundleRelationAttributeMatchingHandler.bundleRelation`,
`CoreBundleRelationAttributeMatchingHandler.bundleRelationAttributeSelectItems`,
`CoreBundleRelationAttributeMatchingHandler.bundleSelectItemsGenerated` und
`CoreBundleRelationAttributeMatchingHandler.password`.

EAMRelation getCoreRelation ()

Definiert in Zeile 849 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Benutzt `CoreBundleRelationAttributeMatchingHandler.coreRelation`.

ArrayList<SelectItem> getCoreRelationAttributeSelectItems ()

Generiert die `SelectItems` der Relations-Attribute einer Kern-Relation für die `SelectListbox` der View. Dabei wird zusätzlich geprüft, ob zu einem Attribut bereits ein Eintrag in der Verknüpfungs-Tabelle der Relations-Attribute vorhanden ist. Ist dies der Fall, wird für dieses Attribut kein `SelectItem` erstellt.

Rückgabe:

Die vollständig generierte Liste der `SelectItems`, eine leere Liste im Fehlerfall.

Definiert in Zeile 659 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Benutzt `CoreBundleRelationAttributeMatchingHandler.coreRelation`,
`CoreBundleRelationAttributeMatchingHandler.coreRelationAttributeSelectItems`,
`CoreBundleRelationAttributeMatchingHandler.coreSelectItemsGenerated` und
`CoreBundleRelationAttributeMatchingHandler.password`.

String getSelectedBundleRelationAttribute ()

Definiert in Zeile 880 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Benutzt `CoreBundleRelationAttributeMatchingHandler.selectedBundleRelationAttribute`.

ArrayList<String> getSelectedBundleRelationAttributes ()

Definiert in Zeile 838 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Benutzt `CoreBundleRelationAttributeMatchingHandler.selectedBundleRelationAttributes`.

String getSelectedCoreRelationAttribute ()

Definiert in Zeile 869 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Benutzt `CoreBundleRelationAttributeMatchingHandler.selectedCoreRelationAttribute`.

void goToMappingPage ()

Setzt die Flags für die Mapping-Page der Relations-Attribute.

Definiert in Zeile 830 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Benutzt `CoreBundleRelationAttributeMatchingHandler.attributeMappingPage` und
`CoreBundleRelationAttributeMatchingHandler.attributeStoringPage`.

void goToStoringPage ()

Setzt die Flags für die Storing-Page der Relations-Attribute.

Definiert in Zeile 821 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Benutzt `CoreBundleRelationAttributeMatchingHandler.attributeMappingPage` und
`CoreBundleRelationAttributeMatchingHandler.attributeStoringPage`.

boolean isAttributeMappingPage ()

Definiert in Zeile 975 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Benutzt `CoreBundleRelationAttributeMatchingHandler.attributeMappingPage`.

boolean isAttributeMatchesEmpty ()

Überprüft, ob die Liste der AttributeMatches leer ist oder nicht.

Rückgabe:

True, wenn die Liste der AttributeMatches nicht leer ist, sonst false.

Definiert in Zeile 554 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.attributeMatches und
CoreBundleRelationAttributeMatchingHandler.attributeMatchesEmpty.

static boolean isAttributeMatchesGenerated () [static]

Definiert in Zeile 912 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.attributeMatchesGenerated.

boolean isAttributeStoringPage ()

Definiert in Zeile 985 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.attributeStoringPage.

static boolean isBundleSelectItemsGenerated () [static]

Definiert in Zeile 891 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.bundleSelectItemsGenerated.

static boolean isCoreSelectItemsGenerated () [static]

Definiert in Zeile 902 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.coreSelectItemsGenerated.

boolean isDeleteMappedAndStored ()

Definiert in Zeile 965 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.deleteMappedAndStored.

boolean isDeleteMappedAttribute ()

Definiert in Zeile 933 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.deleteMappedAttribute.

boolean isDeleteMappedAttributeSelected ()

Definiert in Zeile 954 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.deleteMappedAttributeSelected.

boolean isDeleteStoredAttribute ()

Definiert in Zeile 923 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.deleteStoredAttribute.

boolean isDeleteStoredAttributeSelected ()

Definiert in Zeile 943 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.deleteStoredAttributeSelected.

boolean isSizeGreaterTen ()

Überprüft ob die Liste der AttributeMatches größer 10 ist.

Rückgabe:

True, wenn die Liste der AttributeMatches > 10, sonst false.

Definiert in Zeile 569 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.attributeMatches.

void noDelete ()

Setzt alle Boolean-Flags, die beim Löschen verwendet werden, wieder auf ihren Standard-Wert zurück.

Rückgabe:

Definiert in Zeile 808 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.deleteMappedAndStored,
CoreBundleRelationAttributeMatchingHandler.deleteMappedAttribute,
CoreBundleRelationAttributeMatchingHandler.deleteMappedAttributeSelected,
CoreBundleRelationAttributeMatchingHandler.deleteStoredAttribute und
CoreBundleRelationAttributeMatchingHandler.deleteStoredAttributeSelected.

void resetFlags ()

Setzt alle Boolean-Flags auf ihren Standard-Wert zurück.

Definiert in Zeile 77 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.attributeMatchesEmpty,
CoreBundleRelationAttributeMatchingHandler.attributeMatchesGenerated,
CoreBundleRelationAttributeMatchingHandler.bundleSelectItemsGenerated,
CoreBundleRelationAttributeMatchingHandler.coreSelectItemsGenerated,
CoreBundleRelationAttributeMatchingHandler.deleteMappedAndStored,
CoreBundleRelationAttributeMatchingHandler.deleteMappedAttribute,

CoreBundleRelationAttributeMatchingHandler.deleteMappedAttributeSelected,
CoreBundleRelationAttributeMatchingHandler.deleteStoredAttribute und
CoreBundleRelationAttributeMatchingHandler.deleteStoredAttributeSelected.

Wird benutzt von CoreBundleRelationAttributeMatchingHandler.CoreBundleRelationAttributeMatchingHandler().

void saveAttributeAssignment ()

Speichert die Verknüpfung von Attributen der Kernrelation und einer Relation eines Bundles. Die Einträge der Attribute werden aus ihren jeweiligen Listboxen entfernt und ein Eintrag in AttributeMatches hinzugefügt, um diese in der Datatable anzuzeigen.

Definiert in Zeile 307 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.attributeMatches,
CoreBundleRelationAttributeMatchingHandler.attributeMatchesEmpty,
CoreBundleRelationAttributeMatchingHandler.bundleAttributeId,
CoreBundleRelationAttributeMatchingHandler.bundleRelation,
CoreBundleRelationAttributeMatchingHandler.bundleRelationAttributeSelectItems,
CoreBundleRelationAttributeMatchingHandler.coreAttributeId,
CoreBundleRelationAttributeMatchingHandler.coreRelation,
CoreBundleRelationAttributeMatchingHandler.coreRelationAttributeSelectItems,
CoreBundleRelationAttributeMatchingHandler.password,
CoreBundleRelationAttributeMatchingHandler.selectedBundleRelationAttribute und
CoreBundleRelationAttributeMatchingHandler.selectedCoreRelationAttribute.

static void setAttributeMappingPage (boolean attributeMappingPage) [static]

Definiert in Zeile 980 der Datei CoreBundleRelationAttributeMatchingHandler.java.

static void setAttributeMatchesEmpty (boolean attributeMatchesEmpty) [static]

Definiert in Zeile 995 der Datei CoreBundleRelationAttributeMatchingHandler.java.

static void setAttributeMatchesGenerated (boolean attributeMatchesGenerated) [static]

Definiert in Zeile 917 der Datei CoreBundleRelationAttributeMatchingHandler.java.

static void setAttributeStoringPage (boolean attributeStoringPage) [static]

Definiert in Zeile 990 der Datei CoreBundleRelationAttributeMatchingHandler.java.

static void setBundleRelation (EAMRelation bundleRelation) [static]

Definiert in Zeile 864 der Datei CoreBundleRelationAttributeMatchingHandler.java.

static void setBundleSelectItemsGenerated (boolean bundleSelectItemsGenerated)
[static]

Definiert in Zeile 896 der Datei CoreBundleRelationAttributeMatchingHandler.java.

static void setCoreRelation (EAMRelation coreRelation) [static]

Definiert in Zeile 854 der Datei CoreBundleRelationAttributeMatchingHandler.java.

static void setCoreSelectItemsGenerated (boolean coreSelectItemsGenerated)
[static]

Definiert in Zeile 907 der Datei CoreBundleRelationAttributeMatchingHandler.java.

void setDeleteMappedAndStored (boolean deleteMappedAndStored)

Definiert in Zeile 970 der Datei CoreBundleRelationAttributeMatchingHandler.java.

void setDeleteMappedAttribute (boolean deleteMappedAttribute)

Definiert in Zeile 938 der Datei CoreBundleRelationAttributeMatchingHandler.java.

void setDeleteMappedAttributeSelected (boolean deleteMappedAttributeSelected)

Definiert in Zeile 959 der Datei CoreBundleRelationAttributeMatchingHandler.java.

void setDeleteStoredAttribute (boolean deleteStoredAttribute)

Definiert in Zeile 928 der Datei CoreBundleRelationAttributeMatchingHandler.java.

void setDeleteStoredAttributeSelected (boolean deleteStoredAttributeSelected)

Definiert in Zeile 948 der Datei CoreBundleRelationAttributeMatchingHandler.java.

void setSelectedBundleRelationAttribute (String selectedBundleRelationAttribute)

Definiert in Zeile 885 der Datei CoreBundleRelationAttributeMatchingHandler.java.

**void setSelectedBundleRelationAttributes (ArrayList< String >
selectedBundleRelationAttributes)**

Definiert in Zeile 843 der Datei CoreBundleRelationAttributeMatchingHandler.java.

void setSelectedCoreRelationAttribute (String selectedCoreRelationAttribute)

Definiert in Zeile 874 der Datei CoreBundleRelationAttributeMatchingHandler.java.

void storeAttributesToRelation ()

Speichert eine ausgewählte Anzahl von Attributen dauerhaft an einer Relation, so dass dieses Attribut zur Relation gehört. Die ausgewählten Attribute werden aus der Liste der Attribute entfernt. Zusätzlich werden die Einträge in AttributeMatches gespeichert, um diese in der Datatable anzeigen zu können.

Definiert in Zeile 394 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Benutzt CoreBundleRelationAttributeMatchingHandler.attributeMatches,
CoreBundleRelationAttributeMatchingHandler.attributeMatchesEmpty,
DAOFactoryStaticClass.attributeTypeDAO,
CoreBundleRelationAttributeMatchingHandler.bundleAttributeId,
CoreBundleRelationAttributeMatchingHandler.bundleRelation,
CoreBundleRelationAttributeMatchingHandler.bundleRelationAttributeSelectItems,
CoreBundleRelationAttributeMatchingHandler.coreRelation,
CoreBundleRelationAttributeMatchingHandler.factory,
CoreBundleRelationAttributeMatchingHandler.password,
DAOFactoryStaticClass.relationDAO und
CoreBundleRelationAttributeMatchingHandler.selectedBundleRelationAttributes.

2.3.4.4 Dokumentation der Datenelemente**boolean attributeMappingPage = true [static, private]**

Definiert in Zeile 58 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Wird benutzt von CoreBundleRelationAttributeMatchingHandler.goToMappingPage(),
CoreBundleRelationAttributeMatchingHandler.goToStoringPage() und
CoreBundleRelationAttributeMatchingHandler.isAttributeMappingPage().

ArrayList<EAMRelationAttributeMatchingStore> attributeMatches [private]

Definiert in Zeile 41 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Wird benutzt von CoreBundleRelationAttributeMatchingHandler.checkDelete(),
CoreBundleRelationAttributeMatchingHandler.checkDeleteMarkedElements(),
CoreBundleRelationAttributeMatchingHandler.deleteAssignment(),
CoreBundleRelationAttributeMatchingHandler.deleteSelectedAssignments(),
CoreBundleRelationAttributeMatchingHandler.getAttributeMatches(),
CoreBundleRelationAttributeMatchingHandler.isAttributeMatchesEmpty(),
CoreBundleRelationAttributeMatchingHandler.isSizeGreaterTen(),
CoreBundleRelationAttributeMatchingHandler.saveAttributeAssignment() und
CoreBundleRelationAttributeMatchingHandler.storeAttributesToRelation().

boolean attributeMatchesEmpty = true [static, private]

Definiert in Zeile 60 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Wird benutzt von `CoreBundleRelationAttributeMatchingHandler.deleteAssignment()`,
`CoreBundleRelationAttributeMatchingHandler.getAttributeMatches()`,
`CoreBundleRelationAttributeMatchingHandler.isAttributeMatchesEmpty()`,
`CoreBundleRelationAttributeMatchingHandler.resetFlags()`,
`CoreBundleRelationAttributeMatchingHandler.saveAttributeAssignment()` und
`CoreBundleRelationAttributeMatchingHandler.storeAttributesToRelation()`.

boolean attributeMatchesGenerated = false [static, private]

Definiert in Zeile 57 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Wird benutzt von `CoreBundleRelationAttributeMatchingHandler.deleteAssignment()`,
`CoreBundleRelationAttributeMatchingHandler.getAttributeMatches()`,
`CoreBundleRelationAttributeMatchingHandler.isAttributeMatchesGenerated()` und
`CoreBundleRelationAttributeMatchingHandler.resetFlags()`.

boolean attributeStoringPage = false [static, private]

Definiert in Zeile 59 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Wird benutzt von `CoreBundleRelationAttributeMatchingHandler.goToMappingPage()`,
`CoreBundleRelationAttributeMatchingHandler.goToStoringPage()` und
`CoreBundleRelationAttributeMatchingHandler.isAttributeStoringPage()`.

String bundleAttributeId [private]

Definiert in Zeile 43 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Wird benutzt von `CoreBundleRelationAttributeMatchingHandler.checkDelete()`,
`CoreBundleRelationAttributeMatchingHandler.checkDeleteMarkedElements()`,
`CoreBundleRelationAttributeMatchingHandler.deleteAssignment()`,
`CoreBundleRelationAttributeMatchingHandler.saveAttributeAssignment()` und
`CoreBundleRelationAttributeMatchingHandler.storeAttributesToRelation()`.

EAMRelation bundleRelation [static, private]

Definiert in Zeile 47 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Wird benutzt von `CoreBundleRelationAttributeMatchingHandler.getAttributeMatches()`,
`CoreBundleRelationAttributeMatchingHandler.getBundleRelation()`,
`CoreBundleRelationAttributeMatchingHandler.getBundleRelationAttributeSelectItems()`,
`CoreBundleRelationAttributeMatchingHandler.saveAttributeAssignment()` und
`CoreBundleRelationAttributeMatchingHandler.storeAttributesToRelation()`.

ArrayList<SelectItem> bundleRelationAttributeSelectItems [private]

Definiert in Zeile 37 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Wird benutzt von

CoreBundleRelationAttributeMatchingHandler.getBundleRelationAttributeSelectItems(),
CoreBundleRelationAttributeMatchingHandler.saveAttributeAssignment() und
CoreBundleRelationAttributeMatchingHandler.storeAttributesToRelation().

boolean bundleSelectItemsGenerated = false [static, private]

Definiert in Zeile 55 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Wird benutzt von CoreBundleRelationAttributeMatchingHandler.deleteAssignment(),
CoreBundleRelationAttributeMatchingHandler.getBundleRelationAttributeSelectItems(),
CoreBundleRelationAttributeMatchingHandler.isBundleSelectItemsGenerated() und
CoreBundleRelationAttributeMatchingHandler.resetFlags().

String coreAttributeId [private]

Definiert in Zeile 44 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Wird benutzt von CoreBundleRelationAttributeMatchingHandler.checkDelete(),
CoreBundleRelationAttributeMatchingHandler.checkDeleteMarkedElements(),
CoreBundleRelationAttributeMatchingHandler.deleteAssignment() und
CoreBundleRelationAttributeMatchingHandler.saveAttributeAssignment().

EAMRelation coreRelation [static, private]

Definiert in Zeile 46 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Wird benutzt von CoreBundleRelationAttributeMatchingHandler.deleteAssignment(),
CoreBundleRelationAttributeMatchingHandler.getAttributeMatches(),
CoreBundleRelationAttributeMatchingHandler.getCoreRelation(),
CoreBundleRelationAttributeMatchingHandler.getCoreRelationAttributeSelectItems(),
CoreBundleRelationAttributeMatchingHandler.saveAttributeAssignment() und
CoreBundleRelationAttributeMatchingHandler.storeAttributesToRelation().

ArrayList<SelectItem> coreRelationAttributeSelectItems [private]

Definiert in Zeile 34 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Wird benutzt von

CoreBundleRelationAttributeMatchingHandler.getCoreRelationAttributeSelectItems()
und CoreBundleRelationAttributeMatchingHandler.saveAttributeAssignment().

boolean coreSelectItemsGenerated = false [static, private]

Definiert in Zeile 56 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Wird benutzt von CoreBundleRelationAttributeMatchingHandler.deleteAssignment(),
CoreBundleRelationAttributeMatchingHandler.getCoreRelationAttributeSelectItems(),

`CoreBundleRelationAttributeMatchingHandler.isCoreSelectItemsGenerated()` und `CoreBundleRelationAttributeMatchingHandler.resetFlags()`.

boolean deleteMappedAndStored = false [private]

Definiert in Zeile 53 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Wird benutzt von

`CoreBundleRelationAttributeMatchingHandler.checkDeleteMarkedElements()`,
`CoreBundleRelationAttributeMatchingHandler.deleteAssignment()`,
`CoreBundleRelationAttributeMatchingHandler.isDeleteMappedAndStored()`,
`CoreBundleRelationAttributeMatchingHandler.noDelete()` und
`CoreBundleRelationAttributeMatchingHandler.resetFlags()`.

boolean deleteMappedAttribute = false [private]

Definiert in Zeile 50 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Wird benutzt von `CoreBundleRelationAttributeMatchingHandler.checkDelete()`,
`CoreBundleRelationAttributeMatchingHandler.deleteAssignment()`,
`CoreBundleRelationAttributeMatchingHandler.isDeleteMappedAttribute()`,
`CoreBundleRelationAttributeMatchingHandler.noDelete()` und
`CoreBundleRelationAttributeMatchingHandler.resetFlags()`.

boolean deleteMappedAttributeSelected = false [private]

Definiert in Zeile 52 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Wird benutzt von

`CoreBundleRelationAttributeMatchingHandler.checkDeleteMarkedElements()`,
`CoreBundleRelationAttributeMatchingHandler.deleteAssignment()`,
`CoreBundleRelationAttributeMatchingHandler.isDeleteMappedAttributeSelected()`,
`CoreBundleRelationAttributeMatchingHandler.noDelete()` und
`CoreBundleRelationAttributeMatchingHandler.resetFlags()`.

boolean deleteStoredAttribute = false [private]

Definiert in Zeile 49 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Wird benutzt von `CoreBundleRelationAttributeMatchingHandler.checkDelete()`,
`CoreBundleRelationAttributeMatchingHandler.deleteAssignment()`,
`CoreBundleRelationAttributeMatchingHandler.isDeleteStoredAttribute()`,
`CoreBundleRelationAttributeMatchingHandler.noDelete()` und
`CoreBundleRelationAttributeMatchingHandler.resetFlags()`.

boolean deleteStoredAttributeSelected = false [private]

Definiert in Zeile 51 der Datei `CoreBundleRelationAttributeMatchingHandler.java`.

Wird benutzt von

CoreBundleRelationAttributeMatchingHandler.checkDeleteMarkedElements(),
CoreBundleRelationAttributeMatchingHandler.deleteAssignment(),
CoreBundleRelationAttributeMatchingHandler.isDeleteStoredAttributeSelected(),
CoreBundleRelationAttributeMatchingHandler.noDelete() und
CoreBundleRelationAttributeMatchingHandler.resetFlags().

**DAOFactoryStaticClass factory = DAOFactoryStaticClass.getInstance(userId,
password, this.getClass().getName()) [private]**

Definiert in Zeile 32 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Wird benutzt von CoreBundleRelationAttributeMatchingHandler.deleteAssignment() und
CoreBundleRelationAttributeMatchingHandler.storeAttributesToRelation().

**String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")**

Definiert in Zeile 30 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Wird benutzt von CoreBundleRelationAttributeMatchingHandler.deleteAssignment(),
CoreBundleRelationAttributeMatchingHandler.getAttributeMatches(),
CoreBundleRelationAttributeMatchingHandler.getBundleRelationAttributeSelectItems(),
CoreBundleRelationAttributeMatchingHandler.getCoreRelationAttributeSelectItems(),
CoreBundleRelationAttributeMatchingHandler.saveAttributeAssignment() und
CoreBundleRelationAttributeMatchingHandler.storeAttributesToRelation().

String selectedBundleRelationAttribute [private]

Definiert in Zeile 38 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Wird benutzt von

CoreBundleRelationAttributeMatchingHandler.getSelectedBundleRelationAttribute() und
CoreBundleRelationAttributeMatchingHandler.saveAttributeAssignment().

ArrayList<String> selectedBundleRelationAttributes [private]

Definiert in Zeile 39 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Wird benutzt von

CoreBundleRelationAttributeMatchingHandler.getSelectedBundleRelationAttributes()
und CoreBundleRelationAttributeMatchingHandler.storeAttributesToRelation().

String selectedCoreRelationAttribute [private]

Definiert in Zeile 35 der Datei CoreBundleRelationAttributeMatchingHandler.java.

Wird benutzt von

CoreBundleRelationAttributeMatchingHandler.getSelectedCoreRelationAttribute() und
CoreBundleRelationAttributeMatchingHandler.saveAttributeAssignment().

String **userId** =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Definiert in Zeile 29 der Datei CoreBundleRelationAttributeMatchingHandler.java.

2.3.5 CoreBundleRelationMatchingHandler Klassenreferenz

2.3.5.1 Ausführliche Beschreibung

ManagedBean für das Mapping der Relationen des Kerns und eines Bundles sowie und die Speicherung der Bundle-Relation in den Kern.

Autor:

Jens Henkel

Definiert in Zeile 33 der Datei CoreBundleRelationMatchingHandler.java.

Öffentliche Methoden

- void checkDelete ()
- void checkDeleteMarkedElements ()
- CoreBundleRelationMatchingHandler ()
- void deleteAssignment ()
- void deleteSelectedAssignment ()
- String editAssignment ()
- ArrayList< SelectItem > getBundleRelationSelectItems ()
- ArrayList< EAMRelation > getCoreRelations ()
- ArrayList< SelectItem > getCoreRelationSelectItems ()
- ArrayList< SelectItem > getMetamodelSelectItems ()
- LinkedList< EAMRelationMatchingStore > getRelationMatches ()
- String getSelectedBundleRelation ()
- ArrayList< String > getSelectedBundleRelations ()
- String getSelectedCoreRelation ()
- ArrayList< String > getSelectedMetamodels ()
- String goToAttributeMatching ()
- boolean isBundleSelectItemsEmpty ()
- boolean isDeleteMappedAndStored ()

- boolean isDeleteMappedRelation ()
- boolean isDeleteMappedRelationSelected ()
- boolean isDeleteStoredRelation ()
- boolean isDeleteStoredRelationSelected ()
- boolean isMetamodelSelectItemsGenerated ()
- boolean isRelationMappingPage ()
- boolean isRelationMatchesEmpty ()
- boolean isRelationStoringPage ()
- boolean isSizeGreaterTen ()
- void noDelete ()
- void resetFlags ()
- void setBundleSelectItemsEmpty (boolean bundleSelectItemsEmpty)
- void setCoreRelations (ArrayList< EAMRelation > coreRelations)
- void setDeleteMappedAndStored (boolean deleteMappedAndStored)
- void setDeleteMappedRelation (boolean deleteMappedRelation)
- void setDeleteMappedRelationSelected (boolean deleteMappedRelationSelected)
- void setDeleteStoredRelation (boolean deleteStoredRelation)
- void setDeleteStoredRelationSelected (boolean deleteStoredRelationSelected)
- void setRelationMatchesEmpty (boolean relationMatchesEmpty)
- void setSelectedBundleRelation (String selectedBundleRelation)
- void setSelectedBundleRelations (ArrayList< String > selectedBundleRelations)
- void setSelectedCoreRelation (String selectedCoreRelation)
- void setSelectedMetamodels (ArrayList< String > selectedMetamodels)
- void storeEAMRelations ()

Öffentliche, statische Methoden

- static LinkedList< EAMRelation > getBundleRelations ()
- static boolean isBundleSelectItemsGenerated ()
- static boolean isCoreSelectItemsGenerated ()
- static boolean isRelationMatchesGenerated ()
- static void setBundleRelations (LinkedList< EAMRelation > bundleRelations)
- static void setBundleSelectItemsGenerated (boolean bundleSelectItemsGenerated)
- static void setCoreSelectItemsGenerated (boolean coreSelectItemsGenerated)
- static void setMetamodelSelectItemsGenerated (boolean metamodelSelectItemsGenerated)
- static void setRelationMappingPage (boolean relationMappingPage)
- static void setRelationMatchesGenerated (boolean relationMatchesGenerated)
- static void setRelationStoringPage (boolean relationStoringPage)

Öffentliche Attribute

- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Private Methoden

- void deleteAssignment (EAMRelationMatchingStore store)
- boolean saveRelationAssignment (String bundleRelationId, Long coreRelationId)

Private Attribute

- ArrayList< EAMCategory > allCategories
- String bundleRelationId
- ArrayList< SelectItem > bundleRelationSelectItems
- boolean bundleSelectItemsEmpty = false
- String coreRelationId
- ArrayList< EAMRelation > coreRelations
- ArrayList< SelectItem > coreRelationSelectItems
- boolean deleteMappedAndStored = false
- boolean deleteMappedRelation = false
- boolean deleteMappedRelationSelected = false
- boolean deleteStoredRelation = false
- boolean deleteStoredRelationSelected = false
- DAOFactoryStaticClass factory = DAOFactoryStaticClass.getInstance(userId, password, this.getClass().getName())
- ArrayList< SelectItem > metamodelSelectItems
- LinkedList< EAMRelationMatchingStore > relationMatches
- boolean relationMatchesEmpty = true
- boolean relationMatchesWorkaround = false
- String selectedBundleRelation
- ArrayList< String > selectedBundleRelations = new ArrayList<String>()
- String selectedCoreRelation
- ArrayList< String > selectedMetamodels = new ArrayList<String>()

Statische private Attribute

- static LinkedList< EAMRelation > bundleRelations

- static boolean bundleSelectItemsGenerated = false
- static boolean coreSelectItemsGenerated = false
- static boolean metamodelSelectItemsGenerated = false
- static boolean relationMappingPage = true
- static boolean relationMatchesGenerated = false
- static boolean relationStoringPage = false

2.3.5.2 Beschreibung der Konstruktoren und Destruktoren

CoreBundleRelationMatchingHandler ()

Konstruktor der Bean. Beim Erzeugen der Bean wird die Liste bereits vorhandener Relations-Verknüpfungen geprüft und erstellt. Zusätzlich werden alle Boolean-Flags auf ihren Standard-Wert zurückgesetzt.

Definiert in Zeile 79 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.getRelationMatches() und CoreBundleRelationMatchingHandler.resetFlags().

2.3.5.3 Dokumentation der Elementfunktionen

void checkDelete ()

Wird ausgeführt, nachdem der Löschen-Button innerhalb der Datatable betätigt wird. Abhängig davon, ob es sich um ein StoredRelation handelt oder nicht, wird ein Boolean-Flag gesetzt, der eine Sicherheitsabfrage in der View anzeigt.

Definiert in Zeile 683 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.bundleRelationId, CoreBundleRelationMatchingHandler.coreRelationId, CoreBundleRelationMatchingHandler.deleteMappedRelation, CoreBundleRelationMatchingHandler.deleteStoredRelation und CoreBundleRelationMatchingHandler.relationMatches.

void checkDeleteMarkedElements ()

Wird ausgeführt, nachdem der Löschen-Button unterhalb der Datatable betätigt wird. Dadurch werden Boolean-Flags gesetzt, die abhängig von den markierten Checkboxes sind. Dadurch wird eine Sicherheitsabfrage in der View dargestellt, wodurch das Löschen fortgesetzt oder abgebrochen werden kann.

Definiert in Zeile 718 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.bundleRelationId, CoreBundleRelationMatchingHandler.coreRelationId, CoreBundleRelationMatchingHandler.deleteMappedAndStored,

CoreBundleRelationMatchingHandler.deleteMappedRelationSelected,
CoreBundleRelationMatchingHandler.deleteStoredRelationSelected und
CoreBundleRelationMatchingHandler.relationMatches.

void deleteAssignment (EAMRelationMatchingStore store) [private]

Entfernt eine Verknüpfung von EAM-Relationen aus der DB und fügt den beiden Listboxen wieder einen Eintrag für das freie EAM-Relationen hinzu.

Parameter:

store Der Datenspeicher, dessen enthaltene Informationen genutzt werden, um die Verknüpfung der Relationen zu löschen und die ListBoxen mit den wieder freien EAM-Relationen zu erweitern.

Definiert in Zeile 521 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.bundleRelationSelectItems,
CoreBundleRelationMatchingHandler.bundleSelectItemsEmpty,
CoreBundleRelationMatchingHandler.bundleSelectItemsGenerated,
CoreBundleRelationMatchingHandler.coreRelations,
CoreBundleRelationMatchingHandler.coreSelectItemsGenerated,
CoreBundleRelationMatchingHandler.deleteMappedAndStored,
CoreBundleRelationMatchingHandler.deleteMappedRelation,
CoreBundleRelationMatchingHandler.deleteMappedRelationSelected,
CoreBundleRelationMatchingHandler.deleteStoredRelation,
CoreBundleRelationMatchingHandler.deleteStoredRelationSelected,
CoreBundleRelationMatchingHandler.factory,
CoreBundleRelationMatchingHandler.password, DAOFactoryStaticClass.relationDAO,
CoreBundleRelationMatchingHandler.relationMatches und
CoreBundleRelationMatchingHandler.relationMatchesEmpty.

void deleteAssignment ()

Löscht die Verknüpfung zweier EAM-Relationen aus der Datenbank und entfernt anschließend den Eintrag aus der Liste der Relation-Matches. Des Weiteren werden die Select-Listen um die jeweiligen Relations-Einträge erweitert.

Definiert in Zeile 485 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.bundleRelationId,
CoreBundleRelationMatchingHandler.coreRelationId und
CoreBundleRelationMatchingHandler.relationMatches.

Wird benutzt von CoreBundleRelationMatchingHandler.deleteSelectedAssignment().

void deleteSelectedAssignment ()

Entfernt alle durch eine Checkbox in der Datatable markierten Relations-Verknüpfungen.

Definiert in Zeile 502 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.deleteAssignment() und CoreBundleRelationMatchingHandler.relationMatches.

String editAssignment ()

Wird durch betätigen des Editieren-Buttons in der View ausgeführt. Dadurch werden zwei bereits verknüpfte Kern- und Bundle-Relationen ausgewählt und es wird auf die Seite der Attribut-Verknüpfungen von Relationen weitergeleitet.

Rückgabe:

Der String "relationAttributes", wenn beide Relationen gefunden werden konnten.
Dadurch wird auf die Seite der Attribut-Verknüpfung von Relationen weitergeleitet.
Ein Leerer String im Fehlerfall, wodurch die aktuelle Seite nicht verlassen wird.

Definiert in Zeile 624 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.bundleRelations,
CoreBundleRelationMatchingHandler.coreRelationId,
CoreBundleRelationMatchingHandler.factory und DAOFactoryStaticClass.relationDAO.

static LinkedList<EAMRelation> getBundleRelations () [static]

Definiert in Zeile 1104 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.bundleRelations.

ArrayList<SelectItem> getBundleRelationSelectItems ()

Erstellt für die Relationen des Bundles eine Liste von SelectItems, die in der ListBox verwendet wird. Durch ein Boolean-Flag wird geprüft, ob diese Liste bereits erstellt wurde. Ist das der Fall, wird die Liste nur zurückgegeben, ohne die Liste erneut erstellen zu müssen.

Rückgabe:

Eine Liste mit SelectItems für die Relationen des Bundles. Eine leere Liste im Fehlerfall.

Definiert in Zeile 264 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.bundleRelationId,
CoreBundleRelationMatchingHandler.bundleRelations,
CoreBundleRelationMatchingHandler.bundleRelationSelectItems,
CoreBundleRelationMatchingHandler.bundleSelectItemsEmpty,
CoreBundleRelationMatchingHandler.bundleSelectItemsGenerated,
CoreBundleRelationMatchingHandler.getRelationMatches(),
CoreBundleRelationMatchingHandler.password und
CoreBundleRelationMatchingHandler.relationMatchesWorkaround.

ArrayList<EAMRelation> getCoreRelations ()

Definiert in Zeile 1094 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.coreRelations.

ArrayList<SelectedItem> getCoreRelationSelectItems ()

Erstellt für die Relationen des Kerns eine Liste von SelectItems, die in der ListBox verwendet wird. Durch ein Boolean-Flag wird geprüft, ob diese Liste bereits erstellt wurde. Ist das der Fall, wird die Liste nur zurückgegeben, ohne die Liste erneut erstellen zu müssen.

Rückgabe:

Eine Liste mit SelectItems für die Relationen des Kdorns. Eine leere Liste im Fehlerfall.

Definiert in Zeile 199 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.coreRelations,
CoreBundleRelationMatchingHandler.coreRelationSelectItems,
CoreBundleRelationMatchingHandler.coreSelectItemsGenerated,
CoreBundleRelationMatchingHandler.factory,
CoreBundleRelationMatchingHandler.password und DAOFactoryStaticClass.relationDAO.

ArrayList<SelectedItem> getMetamodelSelectItems ()

Generiert die für die SelectManyBox in object_matching.jsf die SelectItems für die Metamodelle.

Rückgabe:

Die Liste mit allen SelectItems für Metamodelle.

Definiert in Zeile 791 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.allCategories,
DAOFactoryStaticClass.categoryDAO, CoreBundleRelationMatchingHandler.factory,
CoreBundleRelationMatchingHandler.metamodelSelectItems und
CoreBundleRelationMatchingHandler.metamodelSelectItemsGenerated.

LinkedList<EAMRelationMatchingStore> getRelationMatches ()

Ermittelt alle bereits vorhandenen Relations-Verknüpfungen des Bundles aus der Datenbank und speichert diese in einer Liste.

Rückgabe:

Die Liste der verknüpften EAM-Relationen, eine leere Liste im Fehlerfall.

Definiert in Zeile 109 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.bundleRelationId,
CoreBundleRelationMatchingHandler.bundleRelations,
CoreBundleRelationMatchingHandler.factory,
CoreBundleRelationMatchingHandler.password, DAOFactoryStaticClass.relationDAO,
CoreBundleRelationMatchingHandler.relationMatches,
CoreBundleRelationMatchingHandler.relationMatchesEmpty und
CoreBundleRelationMatchingHandler.relationMatchesGenerated.

Wird benutzt von

CoreBundleRelationMatchingHandler.CoreBundleRelationMatchingHandler() und
CoreBundleRelationMatchingHandler.getBundleRelationSelectItems().

String getSelectedBundleRelation ()

Definiert in Zeile 1114 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.selectedBundleRelation.

ArrayList<String> getSelectedBundleRelations ()

Definiert in Zeile 1243 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.selectedBundleRelations.

String getSelectedCoreRelation ()

Definiert in Zeile 1124 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.selectedCoreRelation.

ArrayList<String> getSelectedMetamodels ()

Definiert in Zeile 1233 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.selectedMetamodels.

String goToAttributeMatching ()

Übergibt die beiden Relationen, die miteinander verknüpft werden sollen, an die Bean CoreBundleRelationAttributeMatchingHandler. Anschließend wird die Verknüpfung gespeichert und eine Weiterleitung auf die Seite zur Verknüpfung der Attribute durchgeführt. Tritt ein Fehler auf, wird auf der aktuellen Seite verblieben.

Rückgabe:

Definiert in Zeile 345 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.bundleRelationId,
CoreBundleRelationMatchingHandler.bundleRelations,

CoreBundleRelationMatchingHandler.bundleSelectItemsGenerated,
CoreBundleRelationMatchingHandler.coreRelationId,
CoreBundleRelationMatchingHandler.coreSelectItemsGenerated,
CoreBundleRelationMatchingHandler.factory, DAOFactoryStaticClass.relationDAO,
CoreBundleRelationMatchingHandler.saveRelationAssignment(),
CoreBundleRelationMatchingHandler.selectedBundleRelation und
CoreBundleRelationMatchingHandler.selectedCoreRelation.

boolean isBundleSelectItemsEmpty ()

Definiert in Zeile 1165 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.bundleRelationSelectItems und
CoreBundleRelationMatchingHandler.bundleSelectItemsEmpty.

static boolean isBundleSelectItemsGenerated () [static]

Definiert in Zeile 1134 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.bundleSelectItemsGenerated.

static boolean isCoreSelectItemsGenerated () [static]

Definiert in Zeile 1145 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.coreSelectItemsGenerated.

boolean isDeleteMappedAndStored ()

Definiert in Zeile 1264 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.deleteMappedAndStored.

boolean isDeleteMappedRelation ()

Definiert in Zeile 1203 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.deleteMappedRelation.

boolean isDeleteMappedRelationSelected ()

Definiert in Zeile 1284 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.deleteMappedRelationSelected.

boolean isDeleteStoredRelation ()

Definiert in Zeile 1193 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.deleteStoredRelation.

boolean isDeleteStoredRelationSelected ()

Definiert in Zeile 1274 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.deleteStoredRelationSelected.

boolean isMetamodelSelectItemsGenerated ()

Definiert in Zeile 1253 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.metamodelSelectItemsGenerated.

boolean isRelationMappingPage ()

Definiert in Zeile 1213 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.relationMappingPage.

boolean isRelationMatchesEmpty ()

Definiert in Zeile 1179 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.relationMatches und
CoreBundleRelationMatchingHandler.relationMatchesEmpty.

static boolean isRelationMatchesGenerated () [static]

Definiert in Zeile 1155 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.relationMatchesGenerated.

boolean isRelationStoringPage ()

Definiert in Zeile 1223 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.relationStoringPage.

boolean isSizeGreaterTen ()

Überprüft, ob die Größe der Liste der Relations-Verknüpfungen größer als 10 ist.

Rückgabe:

True, wenn RelationMatches größer 10 ist, sonst false.

Definiert in Zeile 669 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.relationMatches.

void noDelete ()

Setzt alle Boolea-Flags zurück, die beim Löschen einer Verknüpfung benötigt werden. Dadurch wird der Vorgang des Löschens abgebrochen.

Rückgabe:

Definiert in Zeile 761 der Datei `CoreBundleRelationMatchingHandler.java`.

Benutzt `CoreBundleRelationMatchingHandler.deleteMappedAndStored`,
`CoreBundleRelationMatchingHandler.deleteMappedRelation`,
`CoreBundleRelationMatchingHandler.deleteMappedRelationSelected`,
`CoreBundleRelationMatchingHandler.deleteStoredRelation` und
`CoreBundleRelationMatchingHandler.deleteStoredRelationSelected`.

void resetFlags ()

Setzt alle Boolean-Flags auf ihren Standard-Wert zurück.

Definiert in Zeile 88 der Datei `CoreBundleRelationMatchingHandler.java`.

Benutzt `CoreBundleRelationMatchingHandler.bundleSelectItemsGenerated`,
`CoreBundleRelationMatchingHandler.coreSelectItemsGenerated`,
`CoreBundleRelationMatchingHandler.deleteMappedAndStored`,
`CoreBundleRelationMatchingHandler.deleteMappedRelation`,
`CoreBundleRelationMatchingHandler.deleteMappedRelationSelected`,
`CoreBundleRelationMatchingHandler.deleteStoredRelation`,
`CoreBundleRelationMatchingHandler.deleteStoredRelationSelected`,
`CoreBundleRelationMatchingHandler.metamodelSelectItemsGenerated`,
`CoreBundleRelationMatchingHandler.relationMatchesEmpty` und
`CoreBundleRelationMatchingHandler.relationMatchesGenerated`.

Wird benutzt von

`CoreBundleRelationMatchingHandler.CoreBundleRelationMatchingHandler()`.

boolean saveRelationAssignment (String bundleRelationId, Long coreRelationId)
[private]

Speichert die Verknüpfung zweier EAM-Relationen in der Datenbank. Dazu werden die jeweiligen IDs der beiden Relationen benötigt. Die jeweiligen Relationen werden aus den beiden Listen entfernt, um sie nicht weiter verwenden zu können. Zusätzlich wird ein Eintrag in die Liste der Relation-Matches eingefügt, um die Verknüpfung der Relationen in der Datatable anzuzeigen.

Parameter:

bundleRelationId Die eindeutige ID der Bundle-Relation.

coreRelationId Die eindeutige ID der Kern-Relation.

Rückgabe:

True, wenn die Verknüpfung gespeichert werden konnte, sonst false.

Definiert in Zeile 409 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.bundleRelationSelectItems,
CoreBundleRelationMatchingHandler.bundleSelectItemsEmpty,
CoreBundleRelationMatchingHandler.coreRelationSelectItems,
CoreBundleRelationMatchingHandler.password,
CoreBundleRelationMatchingHandler.relationMatches und
CoreBundleRelationMatchingHandler.relationMatchesEmpty.

Wird benutzt von CoreBundleRelationMatchingHandler.goToAttributeMatching().

static void setBundleRelations (LinkedList< EAMRelation > bundleRelations)
[static]

Definiert in Zeile 1109 der Datei CoreBundleRelationMatchingHandler.java.

void setBundleSelectItemsEmpty (boolean bundleSelectItemsEmpty)

Definiert in Zeile 1174 der Datei CoreBundleRelationMatchingHandler.java.

static void setBundleSelectItemsGenerated (boolean bundleSelectItemsGenerated)
[static]

Definiert in Zeile 1139 der Datei CoreBundleRelationMatchingHandler.java.

void setCoreRelations (ArrayList< EAMRelation > coreRelations)

Definiert in Zeile 1099 der Datei CoreBundleRelationMatchingHandler.java.

static void setCoreSelectItemsGenerated (boolean coreSelectItemsGenerated)
[static]

Definiert in Zeile 1150 der Datei CoreBundleRelationMatchingHandler.java.

void setDeleteMappedAndStored (boolean deleteMappedAndStored)

Definiert in Zeile 1269 der Datei CoreBundleRelationMatchingHandler.java.

void setDeleteMappedRelation (boolean deleteMappedRelation)

Definiert in Zeile 1208 der Datei CoreBundleRelationMatchingHandler.java.

void setDeleteMappedRelationSelected (boolean deleteMappedRelationSelected)

Definiert in Zeile 1289 der Datei CoreBundleRelationMatchingHandler.java.

void setDeleteStoredRelation (boolean deleteStoredRelation)

Definiert in Zeile 1198 der Datei CoreBundleRelationMatchingHandler.java.

void setDeleteStoredRelationSelected (boolean deleteStoredRelationSelected)

Definiert in Zeile 1279 der Datei CoreBundleRelationMatchingHandler.java.

static void setMetamodelSelectItemsGenerated (boolean metamodelSelectItemsGenerated) [static]

Definiert in Zeile 1258 der Datei CoreBundleRelationMatchingHandler.java.

static void setRelationMappingPage (boolean relationMappingPage) [static]

Definiert in Zeile 1218 der Datei CoreBundleRelationMatchingHandler.java.

void setRelationMatchesEmpty (boolean relationMatchesEmpty)

Definiert in Zeile 1188 der Datei CoreBundleRelationMatchingHandler.java.

static void setRelationMatchesGenerated (boolean relationMatchesGenerated) [static]

Definiert in Zeile 1160 der Datei CoreBundleRelationMatchingHandler.java.

static void setRelationStoringPage (boolean relationStoringPage) [static]

Definiert in Zeile 1228 der Datei CoreBundleRelationMatchingHandler.java.

void setSelectedBundleRelation (String selectedBundleRelation)

Definiert in Zeile 1119 der Datei CoreBundleRelationMatchingHandler.java.

void setSelectedBundleRelations (ArrayList< String > selectedBundleRelations)

Definiert in Zeile 1248 der Datei CoreBundleRelationMatchingHandler.java.

void setSelectedCoreRelation (String selectedCoreRelation)

Definiert in Zeile 1129 der Datei CoreBundleRelationMatchingHandler.java.

void setSelectedMetamodels (ArrayList< String > selectedMetamodels)

Definiert in Zeile 1238 der Datei CoreBundleRelationMatchingHandler.java.

void storeEAMRelations ()

Speichert eine Relation des Bundles in der Datenbank. Dazu müssen die durch die Relation verbundenen EAM-Objekte entweder bereits mit einem Objekt des Kerns verknüpft sein oder selber in der Datenbank gespeichert worden sein. Ist dies nicht der Fall, wird in der View eine Fehlermeldung ausgegeben.

Definiert in Zeile 833 der Datei CoreBundleRelationMatchingHandler.java.

Benutzt CoreBundleRelationMatchingHandler.allCategories,
CoreBundleRelationMatchingHandler.bundleRelationId,
CoreBundleRelationMatchingHandler.bundleRelations,
CoreBundleRelationMatchingHandler.bundleRelationSelectItems,
CoreBundleRelationMatchingHandler.bundleSelectItemsEmpty,
CoreBundleRelationMatchingHandler.factory, DAOFactoryStaticClass.objectDAO,
CoreBundleRelationMatchingHandler.password, DAOFactoryStaticClass.relationDAO,
CoreBundleRelationMatchingHandler.relationMatches,
CoreBundleRelationMatchingHandler.relationMatchesEmpty,
DAOFactoryStaticClass.relationTypeDAO,
CoreBundleRelationMatchingHandler.selectedBundleRelations,
CoreBundleRelationMatchingHandler.selectedMetamodels,
DAOFactoryStaticClass.statusDAO und DAOFactoryStaticClass.timeDAO.

2.3.5.4 Dokumentation der Datenelemente**ArrayList<EAMCategory> allCategories [private]**

Definiert in Zeile 49 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.getMetamodelSelectItems() und CoreBundleRelationMatchingHandler.storeEAMRelations().

String bundleRelationId [private]

Definiert in Zeile 66 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.checkDelete(),
CoreBundleRelationMatchingHandler.checkDeleteMarkedElements(),
CoreBundleRelationMatchingHandler.deleteAssignment(),
CoreBundleRelationMatchingHandler.getBundleRelationSelectItems(),
CoreBundleRelationMatchingHandler.getRelationMatches(),
CoreBundleRelationMatchingHandler.goToAttributeMatching() und
CoreBundleRelationMatchingHandler.storeEAMRelations().

LinkedList<EAMRelation> bundleRelations [static, private]

Definiert in Zeile 41 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.editAssignment(),

CoreBundleRelationMatchingHandler.getBundleRelations(),
CoreBundleRelationMatchingHandler.getBundleRelationSelectItems(),
CoreBundleRelationMatchingHandler.getRelationMatches(),
CoreBundleRelationMatchingHandler.goToAttributeMatching() und
CoreBundleRelationMatchingHandler.storeEAMRelations().

ArrayList<SelectItem> bundleRelationSelectItems [private]

Definiert in Zeile 40 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.deleteAssignment(),
CoreBundleRelationMatchingHandler.getBundleRelationSelectItems(),
CoreBundleRelationMatchingHandler.isBundleSelectItemsEmpty(),
CoreBundleRelationMatchingHandler.saveRelationAssignment() und
CoreBundleRelationMatchingHandler.storeEAMRelations().

boolean bundleSelectItemsEmpty = false [private]

Definiert in Zeile 62 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.deleteAssignment(),
CoreBundleRelationMatchingHandler.getBundleRelationSelectItems(),
CoreBundleRelationMatchingHandler.isBundleSelectItemsEmpty(),
CoreBundleRelationMatchingHandler.saveRelationAssignment() und
CoreBundleRelationMatchingHandler.storeEAMRelations().

boolean bundleSelectItemsGenerated = false [static, private]

Definiert in Zeile 55 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.deleteAssignment(),
CoreBundleRelationMatchingHandler.getBundleRelationSelectItems(),
CoreBundleRelationMatchingHandler.goToAttributeMatching(),
CoreBundleRelationMatchingHandler.isBundleSelectItemsGenerated() und
CoreBundleRelationMatchingHandler.resetFlags().

String coreRelationId [private]

Definiert in Zeile 67 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.checkDelete(),
CoreBundleRelationMatchingHandler.checkDeleteMarkedElements(),
CoreBundleRelationMatchingHandler.deleteAssignment(),
CoreBundleRelationMatchingHandler.editAssignment() und
CoreBundleRelationMatchingHandler.goToAttributeMatching().

ArrayList<EAMRelation> coreRelations [private]

Definiert in Zeile 45 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.deleteAssignment(), CoreBundleRelationMatchingHandler.getCoreRelations() und CoreBundleRelationMatchingHandler.getCoreRelationSelectItems().

ArrayList<SelectItem> coreRelationSelectItems [private]

Definiert in Zeile 44 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.getCoreRelationSelectItems() und CoreBundleRelationMatchingHandler.saveRelationAssignment().

boolean coreSelectItemsGenerated = false [static, private]

Definiert in Zeile 56 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.deleteAssignment(), CoreBundleRelationMatchingHandler.getCoreRelationSelectItems(), CoreBundleRelationMatchingHandler.goToAttributeMatching(), CoreBundleRelationMatchingHandler.isCoreSelectItemsGenerated() und CoreBundleRelationMatchingHandler.resetFlags().

boolean deleteMappedAndStored = false [private]

Definiert in Zeile 70 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.checkDeleteMarkedElements(), CoreBundleRelationMatchingHandler.deleteAssignment(), CoreBundleRelationMatchingHandler.isDeleteMappedAndStored(), CoreBundleRelationMatchingHandler.noDelete() und CoreBundleRelationMatchingHandler.resetFlags().

boolean deleteMappedRelation = false [private]

Definiert in Zeile 69 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.checkDelete(), CoreBundleRelationMatchingHandler.deleteAssignment(), CoreBundleRelationMatchingHandler.isDeleteMappedRelation(), CoreBundleRelationMatchingHandler.noDelete() und CoreBundleRelationMatchingHandler.resetFlags().

boolean deleteMappedRelationSelected = false [private]

Definiert in Zeile 72 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.checkDeleteMarkedElements(), CoreBundleRelationMatchingHandler.deleteAssignment(),

CoreBundleRelationMatchingHandler.isDeleteMappedRelationSelected(),
CoreBundleRelationMatchingHandler.noDelete() und
CoreBundleRelationMatchingHandler.resetFlags().

boolean deleteStoredRelation = false [private]

Definiert in Zeile 68 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.checkDelete(),
CoreBundleRelationMatchingHandler.deleteAssignment(),
CoreBundleRelationMatchingHandler.isDeleteStoredRelationSelected(),
CoreBundleRelationMatchingHandler.noDelete() und
CoreBundleRelationMatchingHandler.resetFlags().

boolean deleteStoredRelationSelected = false [private]

Definiert in Zeile 71 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.checkDeleteMarkedElements(),
CoreBundleRelationMatchingHandler.deleteAssignment(),
CoreBundleRelationMatchingHandler.isDeleteStoredRelationSelected(),
CoreBundleRelationMatchingHandler.noDelete() und
CoreBundleRelationMatchingHandler.resetFlags().

**DAOFactoryStaticClass factory = DAOFactoryStaticClass.getInstance(userId,
password, this.getClass().getName())** [private]

Definiert in Zeile 38 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.deleteAssignment(),
CoreBundleRelationMatchingHandler.editAssignment(),
CoreBundleRelationMatchingHandler.getCoreRelationSelectItems(),
CoreBundleRelationMatchingHandler.getMetamodelSelectItems(),
CoreBundleRelationMatchingHandler.getRelationMatches(),
CoreBundleRelationMatchingHandler.goToAttributeMatching() und
CoreBundleRelationMatchingHandler.storeEAMRelations().

ArrayList<SelectedItem> metamodelSelectItems [private]

Definiert in Zeile 48 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.getMetamodelSelectItems().

boolean metamodelSelectItemsGenerated = false [static, private]

Definiert in Zeile 58 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.getMetamodelSelectItems(),
CoreBundleRelationMatchingHandler.isMetamodelSelectItemsGenerated() und

CoreBundleRelationMatchingHandler.resetFlags().

String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")

Definiert in Zeile 36 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.deleteAssignment(),
CoreBundleRelationMatchingHandler.getBundleRelationSelectItems(),
CoreBundleRelationMatchingHandler.getCoreRelationSelectItems(),
CoreBundleRelationMatchingHandler.getRelationMatches(),
CoreBundleRelationMatchingHandler.saveRelationAssignment() und
CoreBundleRelationMatchingHandler.storeEAMRelations().

boolean relationMappingPage = true [static, private]

Definiert in Zeile 59 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.isRelationMappingPage().

LinkedList<EAMRelationMatchingStore> relationMatches [private]

Definiert in Zeile 53 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.checkDelete(),
CoreBundleRelationMatchingHandler.checkDeleteMarkedElements(),
CoreBundleRelationMatchingHandler.deleteAssignment(),
CoreBundleRelationMatchingHandler.deleteSelectedAssignment(),
CoreBundleRelationMatchingHandler.getRelationMatches(),
CoreBundleRelationMatchingHandler.isRelationMatchesEmpty(),
CoreBundleRelationMatchingHandler.isSizeGreaterTen(),
CoreBundleRelationMatchingHandler.saveRelationAssignment() und
CoreBundleRelationMatchingHandler.storeEAMRelations().

boolean relationMatchesEmpty = true [private]

Definiert in Zeile 63 der Datei CoreBundleRelationMatchingHandler.java.

Wird benutzt von CoreBundleRelationMatchingHandler.deleteAssignment(),
CoreBundleRelationMatchingHandler.getRelationMatches(),
CoreBundleRelationMatchingHandler.isRelationMatchesEmpty(),
CoreBundleRelationMatchingHandler.resetFlags(),
CoreBundleRelationMatchingHandler.saveRelationAssignment() und
CoreBundleRelationMatchingHandler.storeEAMRelations().

boolean relationMatchesGenerated = false [static, private]

Definiert in Zeile 57 der Datei `CoreBundleRelationMatchingHandler.java`.

Wird benutzt von `CoreBundleRelationMatchingHandler.getRelationMatches()`,
`CoreBundleRelationMatchingHandler.isRelationMatchesGenerated()` und
`CoreBundleRelationMatchingHandler.resetFlags()`.

boolean relationMatchesWorkaround = false [private]

Definiert in Zeile 64 der Datei `CoreBundleRelationMatchingHandler.java`.

Wird benutzt von `CoreBundleRelationMatchingHandler.getBundleRelationSelectItems()`.

boolean relationStoringPage = false [static, private]

Definiert in Zeile 60 der Datei `CoreBundleRelationMatchingHandler.java`.

Wird benutzt von `CoreBundleRelationMatchingHandler.isRelationStoringPage()`.

String selectedBundleRelation [private]

Definiert in Zeile 42 der Datei `CoreBundleRelationMatchingHandler.java`.

Wird benutzt von `CoreBundleRelationMatchingHandler.getSelectedBundleRelation()` und
`CoreBundleRelationMatchingHandler.goToAttributeMatching()`.

ArrayList<String> selectedBundleRelations = new ArrayList<String>() [private]

Definiert in Zeile 51 der Datei `CoreBundleRelationMatchingHandler.java`.

Wird benutzt von `CoreBundleRelationMatchingHandler.getSelectedBundleRelations()` und
`CoreBundleRelationMatchingHandler.storeEAMRelations()`.

String selectedCoreRelation [private]

Definiert in Zeile 46 der Datei `CoreBundleRelationMatchingHandler.java`.

Wird benutzt von `CoreBundleRelationMatchingHandler.getSelectedCoreRelation()` und
`CoreBundleRelationMatchingHandler.goToAttributeMatching()`.

ArrayList<String> selectedMetamodels = new ArrayList<String>() [private]

Definiert in Zeile 50 der Datei `CoreBundleRelationMatchingHandler.java`.

Wird benutzt von `CoreBundleRelationMatchingHandler.getSelectedMetamodels()` und
`CoreBundleRelationMatchingHandler.storeEAMRelations()`.

String userId =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Definiert in Zeile 35 der Datei CoreBundleRelationMatchingHandler.java.

2.3.6 CoreBundleViewRoleMatchingHandler Klassenreferenz

2.3.6.1 Ausführliche Beschreibung

ManagedBean für das Mapping der Rollen des Kerns mit den Views, die aus den Bundles in die Datenbank übertragen werden.

Autor:

Jens Henkel

Definiert in Zeile 36 der Datei CoreBundleViewRoleMatchingHandler.java.

Öffentliche Methoden

- void checkDeleteMarkedElements ()
- void checkDeleteSingleElement ()
- CoreBundleViewRoleMatchingHandler ()
- void deleteAssignment ()
- void deleteSingleAssignment ()
- boolean getBundleHasObjects ()
- LinkedList< View > getBundleViews ()
- ArrayList< SelectItem > getBundleViewSelectItems ()
- ArrayList< Role > getCoreRoles ()
- ArrayList< SelectItem > getCoreRoleSelectItems ()
- LinkedList< Method > getErrorMethods ()
- String getSelectedBundleView ()
- ArrayList< String > getSelectedBundleViews ()
- ArrayList< String > getSelectedCoreRoles ()
- ArrayList< SelectItem > getSelectZach ()
- String getViewDescription ()
- HtmlInputHidden getViewId ()
- LinkedList< EAMViewRoleMatchingStore > getViewRoleMatches ()
- String goableViewErrorDetails ()
- boolean isDeleteMarkedElements ()
- boolean isDeleteSingleElement ()
- boolean isErrorMethodsSizeGreaterTen ()
- boolean isShowViewInfos ()
- boolean isSizeGreaterFive ()
- boolean isSizeGreaterTen ()

- boolean isViewCoreMethodErrors ()
- boolean isViewRoleMatchesEmpty ()
- void listenerSelZachFilter (ValueChangeEvent event)
- void noDelete ()
- void resetFlags ()
- void saveAssignment ()
- void setDeleteMarkedElements (boolean deleteMarkedElements)
- void setDeleteSingleElement (boolean deleteSingleElement)
- void setSelectedBundleView (String selectedBundleView)
- void setSelectedBundleViews (ArrayList< String > selectedBundleViews)
- void setSelectedCoreRoles (ArrayList< String > selectedCoreRoles)
- void setSelectedZach (String selectedZach)
- void setViewCoreMethodErrors (boolean viewCoreMethodErrors)
- void setViewId (HtmlInputHidden viewId)
- void showViewDescription (ValueChangeEvent event)

Öffentliche, statische Methoden

- static CoreBundleViewRoleMatchingHandler getInstance ()
- static boolean isBundleViewSelectItemsGenerated ()
- static boolean isCoreRolesSelectItemsGenerated ()
- static boolean isViewRoleMatchesGenerated ()
- static void setBundleHasObjects (boolean bundleHasObjects)
- static void setBundleViews (LinkedList< View > bundleViews)
- static void setBundleViewSelectItemsGenerated (boolean bundleViewSelectItemsGenerated)
- static void setCoreRolesSelectItemsGenerated (boolean coreRolesSelectItemsGenerated)
- static void setErrorInformationsShown (boolean errorInformationsShown)
- static void setViewDescription (String viewDescription)
- static void setViewRoleMatches (LinkedList< EAMViewRoleMatchingStore > viewRoleMatches)
- static void setViewRoleMatchesEmpty (boolean viewRoleMatchesEmpty)
- static void setViewRoleMatchesGenerated (boolean viewRoleMatchesGenerated)

Private Methoden

- void fillZachmanFilter ()
- void showErrorInformation ()

Private Attribute

- ArrayList< SelectItem > bundleViewSelectItems
- int coreRoleId
- ArrayList< Role > coreRoles
- ArrayList< SelectItem > coreRoleSelectItems
- boolean deleteMarkedElements = false
- boolean deleteSingleElement = false
- LinkedList< Method > errorMethods
- String modViewId
- final String noFilter = "noFilter"
- String selectedBundleView
- ArrayList< String > selectedBundleViews
- ArrayList< String > selectedCoreRoles
- String selectedZach
- ArrayList< SelectItem > selectZach
- boolean showViewInfos = false
- boolean viewCoreMethodErrors = false
- HtmlInputHidden viewId

Statische private Attribute

- static boolean bundleHasObjects = false
- static LinkedList< View > bundleViews
- static boolean bundleViewSelectItemsGenerated = false
- static boolean coreRolesSelectItemsGenerated = false
- static boolean errorInformationsShown = true
- static CoreBundleViewRoleMatchingHandler instance
- static String viewDescription
- static LinkedList< EAMViewRoleMatchingStore > viewRoleMatches
- static boolean viewRoleMatchesEmpty = true
- static boolean viewRoleMatchesGenerated = false

2.3.6.2 Beschreibung der Konstruktoren und Destruktoren**CoreBundleViewRoleMatchingHandler ()**

Konstruktor der Bean. Setzt standardmäßig beim ersten Erstellen alle Boolean-Flags zurück und generiert alle ViewRoleMatches und den Zachman-Filter.

Definiert in Zeile 82 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt `CoreBundleViewRoleMatchingHandler.fillZachmanFilter()`,
`CoreBundleViewRoleMatchingHandler.getViewRoleMatches()`,
`CoreBundleViewRoleMatchingHandler.instance` und
`CoreBundleViewRoleMatchingHandler.resetFlags()`.

2.3.6.3 Dokumentation der Elementfunktionen

void checkDeleteMarkedElements ()

Wird aufgerufen, wenn der untere Löschen-Button verwendet wird. Dazu muss eine der View-Role-Matches durch eine Checkbox markiert sein. Dadurch wird eine Sicherheitsabfrage an den Benutzer gestellt, ob wirklich alle ausgewählten Verknüpfungen gelöscht werden sollen.

Definiert in Zeile 625 der Datei `CoreBundleViewRoleMatchingHandler.java`.

Benutzt `CoreBundleViewRoleMatchingHandler.coreRoleId`,
`CoreBundleViewRoleMatchingHandler.deleteMarkedElements` und
`CoreBundleViewRoleMatchingHandler.modViewId`.

void checkDeleteSingleElement ()

Wird aufgerufen, wenn der Löschen-Button innerhalb der Datatable verwendet wird. Dadurch wird eine Sicherheitsabfrage an den Benutzer gestellt, ob die ausgewählte Verknüpfung tatsächlich gelöscht werden soll.

Definiert in Zeile 638 der Datei `CoreBundleViewRoleMatchingHandler.java`.

Benutzt `CoreBundleViewRoleMatchingHandler.coreRoleId`,
`CoreBundleViewRoleMatchingHandler.deleteSingleElement` und
`CoreBundleViewRoleMatchingHandler.modViewId`.

void deleteAssignment ()

Löscht alle durch die Checkbox in der View markierten View-Role-Matches.

Definiert in Zeile 522 der Datei `CoreBundleViewRoleMatchingHandler.java`.

Benutzt `CoreBundleViewRoleMatchingHandler.deleteMarkedElements`,
`CoreBundleViewRoleMatchingHandler.deleteSingleElement`,
`CoreBundleViewRoleMatchingHandler.selectedBundleView`,
`CoreBundleViewRoleMatchingHandler.showViewInfos`,
`CoreBundleViewRoleMatchingHandler.viewDescription` und
`CoreBundleViewRoleMatchingHandler.viewRoleMatches`.

void deleteSingleAssignment ()

Löscht genau einen Eintrag aus der Liste der View-Role-Matches.

Definiert in Zeile 564 der Datei `CoreBundleViewRoleMatchingHandler.java`.

Benutzt CoreBundleViewRoleMatchingHandler.coreRoleId,
CoreBundleViewRoleMatchingHandler.deleteMarkedElements,
CoreBundleViewRoleMatchingHandler.deleteSingleElement,
CoreBundleViewRoleMatchingHandler.modViewId,
CoreBundleViewRoleMatchingHandler.selectedBundleView,
CoreBundleViewRoleMatchingHandler.showViewInfos,
CoreBundleViewRoleMatchingHandler.viewDescription und
CoreBundleViewRoleMatchingHandler.viewRoleMatches.

void fillZachmanFilter () [private]

Setzt die Filter-Liste für die Zachman-Filterung der Views aus den Bundles.

Definiert in Zeile 685 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.noFilter,
CoreBundleViewRoleMatchingHandler.selectedZach und
CoreBundleViewRoleMatchingHandler.selectZach.

Wird benutzt von

CoreBundleViewRoleMatchingHandler.CoreBundleViewRoleMatchingHandler().

boolean getBundleHasObjects ()

Definiert in Zeile 768 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.bundleHasObjects.

LinkedList<View> getBundleViews ()

Definiert in Zeile 845 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.bundleViews.

ArrayList<SelectItem> getBundleViewSelectItems ()

Generiert für die Views des ausgewählten Bundles eine Liste von SelectItems, die in der View in einer Auswahlliste angezeigt werden. Durch ein Boolean-Flag wird kontrolliert, ob die Liste bereits erstellt wurde. Ist dies der Fall, wird die Liste nur noch zurückgegeben.

Rückgabe:

Die Liste der SelectItems.

Definiert in Zeile 183 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.bundleViews,
CoreBundleViewRoleMatchingHandler.bundleViewSelectItems,
CoreBundleViewRoleMatchingHandler.bundleViewSelectItemsGenerated,
CoreBundleViewRoleMatchingHandler.errorInformationsShown,

CoreBundleViewRoleMatchingHandler.showErrorInformation() und
CoreBundleViewRoleMatchingHandler.viewCoreMethodErrors.

ArrayList<Role> getCoreRoles ()

Definiert in Zeile 753 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.coreRoles.

ArrayList<SelectItem> getCoreRoleSelectItems ()

Generiert für die Rollen des Kerns eine Liste von SelectItems, die in der View in einer Auswahlliste angezeigt werden. Durch ein Boolean-Flag wird kontrolliert, ob die Liste bereits erstellt wurde. Ist dies der Fall, wird die Liste nur noch zurückgegeben.

Rückgabe:

Die Liste der SelectItems.

Definiert in Zeile 121 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.coreRoles,
CoreBundleViewRoleMatchingHandler.coreRoleSelectItems und
CoreBundleViewRoleMatchingHandler.coreRolesSelectItemsGenerated.

LinkedList<Method> getErrorMethods ()

Definiert in Zeile 904 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.errorMethods.

static CoreBundleViewRoleMatchingHandler getInstance () [static]

Rückgabe:

Eine Instanz dieser Bean.

Definiert in Zeile 109 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.instance.

String getSelectedBundleView ()

Definiert in Zeile 820 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.selectedBundleView.

ArrayList<String> getSelectedBundleViews ()

Definiert in Zeile 763 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.selectedBundleViews.

ArrayList<String> getSelectedCoreRoles ()

Definiert in Zeile 758 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.selectedCoreRoles.

ArrayList<SelectedItem> getSelectZach ()

Definiert in Zeile 897 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.selectZach.

String getViewDescription ()

Definiert in Zeile 835 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.viewDescription.

HtmlInputHidden getViewId ()

Definiert in Zeile 919 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.viewId.

LinkedList<EAMViewRoleMatchingStore> getViewRoleMatches ()

Ermittelt alle für das ausgewählte Bundle vorhandenen View- und Rollen-Verknüpfungen und erstellt daraus eine Liste. Dabei wird durch ein Boolean-Flag kontrolliert, ob die Liste bereits erstellt wurde. Ist dies der Fall, wird die Liste nur noch zurückgegeben.

Rückgabe:

Die Liste der View- und Rollen-Verknüpfungen.

Definiert in Zeile 363 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.bundleViews,
CoreBundleViewRoleMatchingHandler.coreRoles,
CoreBundleViewRoleMatchingHandler.viewRoleMatches,
CoreBundleViewRoleMatchingHandler.viewRoleMatchesEmpty und
CoreBundleViewRoleMatchingHandler.viewRoleMatchesGenerated.

Wird benutzt von

CoreBundleViewRoleMatchingHandler.CoreBundleViewRoleMatchingHandler() und
CoreBundleViewRoleMatchingHandler.isViewRoleMatchesEmpty().

String goableViewErrorDetails ()

Ermittelt eine Liste der fehlerhaften Kernmethode, die in der XML-Datei der Views eingegeben wurden. Diese Fehler beziehen sich entweder auf Schreibfehler oder auf veraltete Methode des Kerns, die nicht mehr verwendet werden.

Rückgabe:

Der String der Seite, die als nächstes angezeigt werden soll. Soll zur Fehlerseite gesprungen werden, ist dies "errors". Ansonsten ein leerer String, so das auf der aktuellen Seite verblieben wird.

Definiert in Zeile 711 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.errorMethods und CoreBundleViewRoleMatchingHandler.viewId.

static boolean isBundleViewSelectItemsGenerated () [static]

Definiert in Zeile 788 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.bundleViewSelectItemsGenerated.

static boolean isCoreRolesSelectItemsGenerated () [static]

Definiert in Zeile 799 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.coreRolesSelectItemsGenerated.

boolean isDeleteMarkedElements ()

Definiert in Zeile 855 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.deleteMarkedElements.

boolean isDeleteSingleElement ()

Definiert in Zeile 865 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.deleteSingleElement.

boolean isErrorMethodsSizeGreaterTen ()

Überprüft, ob die Liste der Fehler der View-Kernmethoden-Verknüpfungen größer 10 ist.

Rückgabe:

True, wenn die Liste größer 10 ist, sonst false.

Definiert in Zeile 742 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.errorMethods.

boolean isShowViewInfos ()

Definiert in Zeile 830 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.showViewInfos.

boolean isSizeGreaterFive ()

Überprüft, ob die Größe der View-Role-Matches-Liste größer 10 ist.

Rückgabe:

True, wenn die Liste größer 10 ist, sonst false.

Definiert in Zeile 444 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.viewRoleMatches.

boolean isSizeGreaterTen ()

Überprüft, ob die Größe der Bundle-Views-Liste größer 5 ist.

Rückgabe:

True, wenn die Liste der Bundle-Views größer 5 ist, sonst false.

Definiert in Zeile 460 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.bundleViews.

boolean isViewCoreMethodErrors ()

Definiert in Zeile 909 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.viewCoreMethodErrors.

boolean isViewRoleMatchesEmpty ()

Definiert in Zeile 875 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.getViewRoleMatches(),
CoreBundleViewRoleMatchingHandler.viewRoleMatches und
CoreBundleViewRoleMatchingHandler.viewRoleMatchesEmpty.

static boolean isViewRoleMatchesGenerated () [static]

Definiert in Zeile 810 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.viewRoleMatchesGenerated.

void listenerSelZachFilter (ValueChangeEvent event)

Listener, der auf die Auswahl eines Zachman-Wertes reagiert. Filtert die Selectitems, die die Sichten des Bundles repräsentieren.

Parameter:

event,der Event mit den notwendigen Infos

Definiert in Zeile 662 der Datei `CoreBundleViewRoleMatchingHandler.java`.

Benutzt `CoreBundleViewRoleMatchingHandler.bundleViews`,
`CoreBundleViewRoleMatchingHandler.bundleViewSelectItems`,
`CoreBundleViewRoleMatchingHandler.noFilter` und
`CoreBundleViewRoleMatchingHandler.selectedZach`.

void noDelete ()

Setzt die Sicherheitsabfragen für das Löschen wieder zurück und bricht den Löschen-Vorgang ab.

Definiert in Zeile 650 der Datei `CoreBundleViewRoleMatchingHandler.java`.

Benutzt `CoreBundleViewRoleMatchingHandler.deleteMarkedElements` und
`CoreBundleViewRoleMatchingHandler.deleteSingleElement`.

void resetFlags ()

Setzt alle Boolean-Flags zurück.

Definiert in Zeile 93 der Datei `CoreBundleViewRoleMatchingHandler.java`.

Benutzt `CoreBundleViewRoleMatchingHandler.bundleViewSelectItemsGenerated`,
`CoreBundleViewRoleMatchingHandler.coreRolesSelectItemsGenerated`,
`CoreBundleViewRoleMatchingHandler.deleteMarkedElements`,
`CoreBundleViewRoleMatchingHandler.deleteSingleElement`,
`CoreBundleViewRoleMatchingHandler.showViewInfos`,
`CoreBundleViewRoleMatchingHandler.viewCoreMethodErrors`,
`CoreBundleViewRoleMatchingHandler.viewRoleMatchesEmpty` und
`CoreBundleViewRoleMatchingHandler.viewRoleMatchesGenerated`.

Wird benutzt von

`CoreBundleViewRoleMatchingHandler.CoreBundleViewRoleMatchingHandler()`.

void saveAssignment ()

Speichert Verknüpfungen von Views mit einer Rolle des Kerns. Dabei können mehrere Rollen ausgewählt sein. Zusätzlich wird geprüft, ob diese Zuweisung bereits existiert. Ist dies der Fall, wird diese Zuweisung übersprungen.

Definiert in Zeile 242 der Datei `CoreBundleViewRoleMatchingHandler.java`.

Benutzt `CoreBundleViewRoleMatchingHandler.bundleViews`,
`CoreBundleViewRoleMatchingHandler.coreRoles`,
`CoreBundleViewRoleMatchingHandler.selectedBundleView`,
`CoreBundleViewRoleMatchingHandler.selectedCoreRoles`,
`CoreBundleViewRoleMatchingHandler.showViewInfos`,
`CoreBundleViewRoleMatchingHandler.viewDescription`,
`CoreBundleViewRoleMatchingHandler.viewRoleMatches` und

CoreBundleViewRoleMatchingHandler.viewRoleMatchesGenerated.

static void setBundleHasObjects (boolean bundleHasObjects) [static]

Definiert in Zeile 773 der Datei CoreBundleViewRoleMatchingHandler.java.

static void setBundleViews (LinkedList< View > bundleViews) [static]

Definiert in Zeile 850 der Datei CoreBundleViewRoleMatchingHandler.java.

static void setBundleViewSelectItemsGenerated (boolean bundleViewSelectItemsGenerated) [static]

Definiert in Zeile 793 der Datei CoreBundleViewRoleMatchingHandler.java.

static void setCoreRolesSelectItemsGenerated (boolean coreRolesSelectItemsGenerated) [static]

Definiert in Zeile 804 der Datei CoreBundleViewRoleMatchingHandler.java.

void setDeleteMarkedElements (boolean deleteMarkedElements)

Definiert in Zeile 860 der Datei CoreBundleViewRoleMatchingHandler.java.

void setDeleteSingleElement (boolean deleteSingleElement)

Definiert in Zeile 870 der Datei CoreBundleViewRoleMatchingHandler.java.

static void setErrorInformationsShown (boolean errorInformationsShown) [static]

Definiert in Zeile 943 der Datei CoreBundleViewRoleMatchingHandler.java.

void setSelectedBundleView (String selectedBundleView)

Definiert in Zeile 825 der Datei CoreBundleViewRoleMatchingHandler.java.

void setSelectedBundleViews (ArrayList< String > selectedBundleViews)

Definiert in Zeile 783 der Datei CoreBundleViewRoleMatchingHandler.java.

void setSelectedCoreRoles (ArrayList< String > selectedCoreRoles)

Definiert in Zeile 778 der Datei CoreBundleViewRoleMatchingHandler.java.

void setSelectedZach (String selectedZach)

Definiert in Zeile 900 der Datei CoreBundleViewRoleMatchingHandler.java.

void setViewCoreMethodErrors (boolean viewCoreMethodErrors)

Definiert in Zeile 914 der Datei CoreBundleViewRoleMatchingHandler.java.

static void setViewDescription (String viewDescription) [static]

Definiert in Zeile 840 der Datei CoreBundleViewRoleMatchingHandler.java.

void setViewId (HtmlInputHidden viewId)

Definiert in Zeile 924 der Datei CoreBundleViewRoleMatchingHandler.java.

static void setViewRoleMatches (LinkedList< EAMViewRoleMatchingStore > viewRoleMatches) [static]

Definiert in Zeile 891 der Datei CoreBundleViewRoleMatchingHandler.java.

static void setViewRoleMatchesEmpty (boolean viewRoleMatchesEmpty) [static]

Definiert in Zeile 886 der Datei CoreBundleViewRoleMatchingHandler.java.

static void setViewRoleMatchesGenerated (boolean viewRoleMatchesGenerated) [static]

Definiert in Zeile 815 der Datei CoreBundleViewRoleMatchingHandler.java.

void showErrorInformation () [private]

Definiert in Zeile 929 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.viewCoreMethodErrors.

Wird benutzt von CoreBundleViewRoleMatchingHandler.getBundleViewSelectItems().

void showViewDescription (ValueChangeEvent event)

Listener-Methode, die auf die Auswahl einer View in der Select-Listbox der Views eines Bundles reagiert. Dadurch wird die Beschreibung der View auf der linken Seite eingeblendet.

Parameter:

event Der Event mit den benötigten Infos

Definiert in Zeile 477 der Datei CoreBundleViewRoleMatchingHandler.java.

Benutzt CoreBundleViewRoleMatchingHandler.selectedBundleView, CoreBundleViewRoleMatchingHandler.showViewInfos und CoreBundleViewRoleMatchingHandler.viewDescription.

2.3.6.4 Dokumentation der Datenelemente

boolean bundleHasObjects = false [static, private]

Definiert in Zeile 60 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.getBundleHasObjects().

LinkedList<View> bundleViews [static, private]

Definiert in Zeile 47 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.getBundleViews(),
CoreBundleViewRoleMatchingHandler.getBundleViewSelectItems(),
CoreBundleViewRoleMatchingHandler.getViewRoleMatches(),
CoreBundleViewRoleMatchingHandler.isSizeGreaterTen(),
CoreBundleViewRoleMatchingHandler.listenerSelZachFilter() und
CoreBundleViewRoleMatchingHandler.saveAssignment().

ArrayList<SelectItem> bundleViewSelectItems [private]

Definiert in Zeile 44 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.getBundleViewSelectItems() und
CoreBundleViewRoleMatchingHandler.listenerSelZachFilter().

boolean bundleViewSelectItemsGenerated = false [static, private]

Definiert in Zeile 57 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.getBundleViewSelectItems(),
CoreBundleViewRoleMatchingHandler.isBundleViewSelectItemsGenerated() und
CoreBundleViewRoleMatchingHandler.resetFlags().

int coreRoleId [private]

Definiert in Zeile 74 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.checkDeleteMarkedElements(),
CoreBundleViewRoleMatchingHandler.checkDeleteSingleElement() und
CoreBundleViewRoleMatchingHandler.deleteSingleAssignment().

ArrayList<Role> coreRoles [private]

Definiert in Zeile 41 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.getCoreRoles(),
CoreBundleViewRoleMatchingHandler.getCoreRoleSelectItems(),
CoreBundleViewRoleMatchingHandler.getViewRoleMatches() und
CoreBundleViewRoleMatchingHandler.saveAssignment().

ArrayList<SelectedItem> coreRoleSelectItems [private]

Definiert in Zeile 40 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.getCoreRoleSelectItems().

boolean coreRolesSelectItemsGenerated = false [static, private]

Definiert in Zeile 58 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.getCoreRoleSelectItems(),
CoreBundleViewRoleMatchingHandler.isCoreRolesSelectItemsGenerated() und
CoreBundleViewRoleMatchingHandler.resetFlags().

boolean deleteMarkedElements = false [private]

Definiert in Zeile 64 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.checkDeleteMarkedElements(),
CoreBundleViewRoleMatchingHandler.deleteAssignment(),
CoreBundleViewRoleMatchingHandler.deleteSingleAssignment(),
CoreBundleViewRoleMatchingHandler.isDeleteMarkedElements(),
CoreBundleViewRoleMatchingHandler.noDelete() und
CoreBundleViewRoleMatchingHandler.resetFlags().

boolean deleteSingleElement = false [private]

Definiert in Zeile 65 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.checkDeleteSingleElement(),
CoreBundleViewRoleMatchingHandler.deleteAssignment(),
CoreBundleViewRoleMatchingHandler.deleteSingleAssignment(),
CoreBundleViewRoleMatchingHandler.isDeleteSingleElement(),
CoreBundleViewRoleMatchingHandler.noDelete() und
CoreBundleViewRoleMatchingHandler.resetFlags().

boolean errorInformationsShown = true [static, private]

Definiert in Zeile 69 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.getBundleViewSelectItems().

LinkedList<Method> errorMethods [private]

Definiert in Zeile 55 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.getErrorMethods(),
CoreBundleViewRoleMatchingHandler.goToViewErrorDetails() und
CoreBundleViewRoleMatchingHandler.isErrorMethodsSizeGreaterTen().

CoreBundleViewRoleMatchingHandler instance [static, private]

Definiert in Zeile 38 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von

CoreBundleViewRoleMatchingHandler.CoreBundleViewRoleMatchingHandler() und
CoreBundleViewRoleMatchingHandler.getInstance().

String modViewId [private]

Definiert in Zeile 73 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.checkDeleteMarkedElements(),
CoreBundleViewRoleMatchingHandler.checkDeleteSingleElement() und
CoreBundleViewRoleMatchingHandler.deleteSingleAssignment().

final String noFilter = "noFilter" [private]

Definiert in Zeile 51 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.fillZachmanFilter() und
CoreBundleViewRoleMatchingHandler.listenerSelZachFilter().

String selectedBundleView [private]

Definiert in Zeile 46 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.deleteAssignment(),
CoreBundleViewRoleMatchingHandler.deleteSingleAssignment(),
CoreBundleViewRoleMatchingHandler.getSelectedBundleView(),
CoreBundleViewRoleMatchingHandler.saveAssignment() und
CoreBundleViewRoleMatchingHandler.showViewDescription().

ArrayList<String> selectedBundleViews [private]

Definiert in Zeile 45 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.getSelectedBundleViews().

ArrayList<String> selectedCoreRoles [private]

Definiert in Zeile 42 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.getSelectedCoreRoles() und
CoreBundleViewRoleMatchingHandler.saveAssignment().

String selectedZach [private]

Definiert in Zeile 50 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von `CoreBundleViewRoleMatchingHandler.fillZachmanFilter()` und `CoreBundleViewRoleMatchingHandler.listenerSelZachFilter()`.

ArrayList<SelectedItem> selectZach [private]

Definiert in Zeile 49 der Datei `CoreBundleViewRoleMatchingHandler.java`.

Wird benutzt von `CoreBundleViewRoleMatchingHandler.fillZachmanFilter()` und `CoreBundleViewRoleMatchingHandler.getSelectedZach()`.

boolean showViewInfos = false [private]

Definiert in Zeile 63 der Datei `CoreBundleViewRoleMatchingHandler.java`.

Wird benutzt von `CoreBundleViewRoleMatchingHandler.deleteAssignment()`, `CoreBundleViewRoleMatchingHandler.deleteSingleAssignment()`, `CoreBundleViewRoleMatchingHandler.isShowViewInfos()`, `CoreBundleViewRoleMatchingHandler.resetFlags()`, `CoreBundleViewRoleMatchingHandler.saveAssignment()` und `CoreBundleViewRoleMatchingHandler.showViewDescription()`.

boolean viewCoreMethodErrors = false [private]

Definiert in Zeile 68 der Datei `CoreBundleViewRoleMatchingHandler.java`.

Wird benutzt von `CoreBundleViewRoleMatchingHandler.getBundleViewSelectItems()`, `CoreBundleViewRoleMatchingHandler.isViewCoreMethodErrors()`, `CoreBundleViewRoleMatchingHandler.resetFlags()` und `CoreBundleViewRoleMatchingHandler.showErrorInformation()`.

String viewDescription [static, private]

Definiert in Zeile 71 der Datei `CoreBundleViewRoleMatchingHandler.java`.

Wird benutzt von `CoreBundleViewRoleMatchingHandler.deleteAssignment()`, `CoreBundleViewRoleMatchingHandler.deleteSingleAssignment()`, `CoreBundleViewRoleMatchingHandler.getViewDescription()`, `CoreBundleViewRoleMatchingHandler.saveAssignment()` und `CoreBundleViewRoleMatchingHandler.showViewDescription()`.

HtmlInputHidden viewId [private]

Definiert in Zeile 76 der Datei `CoreBundleViewRoleMatchingHandler.java`.

Wird benutzt von `CoreBundleViewRoleMatchingHandler.getViewId()` und `CoreBundleViewRoleMatchingHandler.goToViewErrorDetails()`.

LinkedList<EAMViewRoleMatchingStore> viewRoleMatches [static, private]

Definiert in Zeile 53 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.deleteAssignment(),
CoreBundleViewRoleMatchingHandler.deleteSingleAssignment(),
CoreBundleViewRoleMatchingHandler.getViewRoleMatches(),
CoreBundleViewRoleMatchingHandler.isSizeGreaterFive(),
CoreBundleViewRoleMatchingHandler.isViewRoleMatchesEmpty() und
CoreBundleViewRoleMatchingHandler.saveAssignment().

boolean viewRoleMatchesEmpty = true [static, private]

Definiert in Zeile 66 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.getViewRoleMatches(),
CoreBundleViewRoleMatchingHandler.isViewRoleMatchesEmpty() und
CoreBundleViewRoleMatchingHandler.resetFlags().

boolean viewRoleMatchesGenerated = false [static, private]

Definiert in Zeile 59 der Datei CoreBundleViewRoleMatchingHandler.java.

Wird benutzt von CoreBundleViewRoleMatchingHandler.getViewRoleMatches(),
CoreBundleViewRoleMatchingHandler.isViewRoleMatchesGenerated(),
CoreBundleViewRoleMatchingHandler.resetFlags() und
CoreBundleViewRoleMatchingHandler.saveAssignment().

2.3.7 DAOFactoryStaticClass Klassenreferenz

2.3.7.1 Ausführliche Beschreibung

Definiert in Zeile 17 der Datei DAOFactoryStaticClass.java.

Öffentliche, statische Methoden

- static synchronized DAOFactoryStaticClass getInstance (String userId, String password, String clazz)

Öffentliche Attribute

- EAMAttributeDAO attributeDAO
- EAMAttributeTypeDAO attributeTypeDAO
- EAMCategoryDAO categoryDAO
- EAMObjectDAO objectDAO
- EAMObjectTypeDAO objectTypeDAO
- EAMRelationDAO relationDAO

- EAMRelationTypeDAO relationTypeDAO
- EAMStatusDAO statusDAO
- EAMTimeDAO timeDAO

Private Methoden

- DAOFactoryStaticClass (String userId, String password)
- DAOFactoryStaticClass ()

Statische private Attribute

- static HashMap< String, DAOFactoryStaticClass > factories = new HashMap<String, DAOFactoryStaticClass>()
- static HashSet< String > registered = new HashSet<String>()

2.3.7.2 Beschreibung der Konstruktoren und Destruktoren

DAOFactoryStaticClass () [private]

Definiert in Zeile 66 der Datei DAOFactoryStaticClass.java.

Wird benutzt von DAOFactoryStaticClass.getInstance().

DAOFactoryStaticClass (String userId, String password) [private]

Definiert in Zeile 70 der Datei DAOFactoryStaticClass.java.

Benutzt DAOFactoryStaticClass.attributeDAO, DAOFactoryStaticClass.attributeTypeDAO, DAOFactoryStaticClass.categoryDAO, DAOFactoryStaticClass.objectDAO, DAOFactoryStaticClass.objectTypeDAO, DAOFactoryStaticClass.relationDAO, DAOFactoryStaticClass.relationTypeDAO, DAOFactoryStaticClass.statusDAO und DAOFactoryStaticClass.timeDAO.

2.3.7.3 Dokumentation der Elementfunktionen

static synchronized DAOFactoryStaticClass getInstance (String userId, String password, String clazz) [static]

Definiert in Zeile 45 der Datei DAOFactoryStaticClass.java.

Benutzt DAOFactoryStaticClass.DAOFactoryStaticClass(), DAOFactoryStaticClass.factories und DAOFactoryStaticClass.registered.

2.3.7.4 Dokumentation der Datenelemente

EAMAttributeDAO attributeDAO

Definiert in Zeile 22 der Datei DAOFactoryStaticClass.java.

Wird benutzt von DAOFactoryStaticClass.DAOFactoryStaticClass(),
CoreBundleRelationAttributeMatchingHandler.deleteAssignment() und
CoreBundleObjectAttributeMatchingHandler.deleteAssignment().

EAMAttributeTypeDAO attributeTypeDAO

Definiert in Zeile 23 der Datei DAOFactoryStaticClass.java.

Wird benutzt von DAOFactoryStaticClass.DAOFactoryStaticClass(),
CoreBundleObjectAttributeMatchingHandler.storeAttributesToObject() und
CoreBundleRelationAttributeMatchingHandler.storeAttributesToRelation().

EAMCategoryDAO categoryDAO

Definiert in Zeile 27 der Datei DAOFactoryStaticClass.java.

Wird benutzt von DAOFactoryStaticClass.DAOFactoryStaticClass() und
CoreBundleRelationMatchingHandler.getMetamodelSelectItems().

**HashMap<String, DAOFactoryStaticClass> factories = new HashMap<String,
DAOFactoryStaticClass>() [static, private]**

Definiert in Zeile 41 der Datei DAOFactoryStaticClass.java.

Wird benutzt von DAOFactoryStaticClass.getInstance().

EAMObjectDAO objectDAO

Definiert in Zeile 19 der Datei DAOFactoryStaticClass.java.

Wird benutzt von DAOFactoryStaticClass.DAOFactoryStaticClass(),
CoreBundleObjectAttributeMatchingHandler.deleteAssignment(),
BundleManager.deleteWithDependencies(),
CoreBundleObjectMatchingHandler.getCoreObjectSelectItems(),
CoreBundleObjectAttributeMatchingHandler.storeAttributesToObject() und
CoreBundleRelationMatchingHandler.storeEAMRelations().

EAMObjectTypeDAO objectTypeDAO

Definiert in Zeile 20 der Datei DAOFactoryStaticClass.java.

Wird benutzt von DAOFactoryStaticClass.DAOFactoryStaticClass().

HashSet<String> registered = new HashSet<String>() [static, private]

Definiert in Zeile 36 der Datei DAOFactoryStaticClass.java.

Wird benutzt von DAOFactoryStaticClass.getInstance().

EAMRelationDAO relationDAO

Definiert in Zeile 29 der Datei DAOFactoryStaticClass.java.

Wird benutzt von DAOFactoryStaticClass.DAOFactoryStaticClass(),
CoreBundleRelationMatchingHandler.deleteAssignment(),
CoreBundleRelationAttributeMatchingHandler.deleteAssignment(),
BundleManager.deleteWithDependencies(),
CoreBundleRelationMatchingHandler.editAssignment(),
CoreBundleRelationMatchingHandler.getCoreRelationSelectItems(),
CoreBundleRelationMatchingHandler.getRelationMatches(),
CoreBundleRelationMatchingHandler.goToAttributeMatching(),
CoreBundleRelationAttributeMatchingHandler.storeAttributesToRelation() und
CoreBundleRelationMatchingHandler.storeEAMRelations().

EAMRelationTypeDAO relationTypeDAO

Definiert in Zeile 30 der Datei DAOFactoryStaticClass.java.

Wird benutzt von DAOFactoryStaticClass.DAOFactoryStaticClass() und
CoreBundleRelationMatchingHandler.storeEAMRelations().

EAMStatusDAO statusDAO

Definiert in Zeile 25 der Datei DAOFactoryStaticClass.java.

Wird benutzt von DAOFactoryStaticClass.DAOFactoryStaticClass() und
CoreBundleRelationMatchingHandler.storeEAMRelations().

EAMTimeDAO timeDAO

Definiert in Zeile 26 der Datei DAOFactoryStaticClass.java.

Wird benutzt von DAOFactoryStaticClass.DAOFactoryStaticClass() und
CoreBundleRelationMatchingHandler.storeEAMRelations().

2.4 Sonstiges

2.4.1 EAMBundle Klassenreferenz

Abgeleitet von org::osgi::framework::Bundle.

2.4.1.1 Ausführliche Beschreibung

Klasse, die die Interface Bundle implementiert und erweitert. Diese Klasse enthält boolesche Methoden, mit denen angegeben wird, ob ein Bundle gestartet oder gestoppt werden kann.

Autor:

Jens Henkel

Definiert in Zeile 22 der Datei EAMBundle.java.

Öffentliche Methoden

- EAMBundle (Bundle x)
- Enumeration findEntries (String arg0, String arg1, boolean arg2)
- BundleContext getBundleContext ()
- long getBundleId ()
- URL getEntry (String arg0)
- Enumeration getEntryPaths (String arg0)
- Dictionary getHeaders (String arg0)
- Dictionary getHeaders ()
- long getLastModified ()
- String getLocation ()
- ServiceReference[] getRegisteredServices ()
- URL getResource (String arg0)
- Enumeration getResources (String arg0) throws IOException
- ServiceReference[] getServicesInUse ()
- int getState ()
- String getSymbolicName ()
- boolean hasPermission (Object arg0)
- boolean isStartable ()
- boolean isStopable ()
- Class loadClass (String arg0) throws ClassNotFoundException
- void start (int arg0) throws BundleException
- void start () throws BundleException
- void stop (int arg0) throws BundleException
- void stop () throws BundleException
- void uninstall () throws BundleException
- void update (InputStream arg0) throws BundleException
- void update () throws BundleException

Private Attribute

- Bundle bundle

2.4.1.2 Beschreibung der Konstruktoren und Destruktoren

EAMBundle (Bundle x)

Definiert in Zeile 27 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

2.4.1.3 Dokumentation der Elementfunktionen

Enumeration findEntries (String arg0, String arg1, boolean arg2)

Definiert in Zeile 34 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

BundleContext getBundleContext ()

Definiert in Zeile 40 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

long getBundleId ()

Definiert in Zeile 46 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

URL getEntry (String arg0)

Definiert in Zeile 52 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

Enumeration getEntryPaths (String arg0)

Definiert in Zeile 58 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

Dictionary getHeaders (String arg0)

Definiert in Zeile 70 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

Dictionary getHeaders ()

Definiert in Zeile 64 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

long getLastModified ()

Definiert in Zeile 76 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

String getLocation ()

Definiert in Zeile 82 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

ServiceReference [] getRegisteredServices ()

Definiert in Zeile 88 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

URL getResource (String arg0)

Definiert in Zeile 94 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

Enumeration getResources (String arg0) throws IOException

Definiert in Zeile 100 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

ServiceReference [] getServicesInUse ()

Definiert in Zeile 106 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

int getState ()

Definiert in Zeile 112 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

Wird benutzt von EAMBundle.isStartable() und EAMBundle.isStopable().

String getSymbolicName ()

Definiert in Zeile 118 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

Wird benutzt von EAMBundle.isStartable() und EAMBundle.isStopable().

boolean hasPermission (Object arg0)

Definiert in Zeile 124 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

boolean isStartable ()

Ermittelt, ob das Bundle gestartet werden kann. Bis auf den Kern und den Bundlemanager können alle Bundles gestartet werden.

Rückgabe:

True, falls das Bundle gestartet werden kann, sonst false.

Definiert in Zeile 183 der Datei EAMBundle.java.

Benutzt EAMBundle.getState() und EAMBundle.getSymbolicName().

boolean isStopable ()

Ermittelt, ob das Bundle gestoppt werden kann. Bis auf den Kern und den Bundlemanager können alle Bundles gestoppt werden.

Rückgabe:

True, falls das Bundle gestoppt werden kann, sonst false.

Definiert in Zeile 209 der Datei EAMBundle.java.

Benutzt EAMBundle.getState() und EAMBundle.getSymbolicName().

Class loadClass (String arg0) throws ClassNotFoundException

Definiert in Zeile 130 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

void start (int arg0) throws BundleException

Definiert in Zeile 142 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

void start () throws BundleException

Definiert in Zeile 136 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

void stop (int arg0) throws BundleException

Definiert in Zeile 154 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

void stop () throws BundleException

Definiert in Zeile 148 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

void uninstall () throws BundleException

Definiert in Zeile 160 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

void update (InputStream arg0) throws BundleException

Definiert in Zeile 172 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

void update () throws BundleException

Definiert in Zeile 166 der Datei EAMBundle.java.

Benutzt EAMBundle.bundle.

2.4.1.4 Dokumentation der Datenelemente

Bundle bundle [private]

Definiert in Zeile 24 der Datei EAMBundle.java.

Wird benutzt von EAMBundle.EAMBundle(), EAMBundle.findEntries(), EAMBundle.getBundleContext(), EAMBundle.getBundleId(), EAMBundle.getEntry(), EAMBundle.getEntryPaths(), EAMBundle.getHeaders(), EAMBundle.getLastModified(), EAMBundle.getLocation(), EAMBundle.getRegisteredServices(), EAMBundle.getResource(), EAMBundle.getResources(), EAMBundle.getServicesInUse(), EAMBundle.getState(), EAMBundle.getSymbolicName(), EAMBundle.hasPermission(), EAMBundle.loadClass(), EAMBundle.start(), EAMBundle.stop(), EAMBundle.uninstall() und EAMBundle.update().

2.4.2 BundleMgrContributor Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::api::AbstractAdoptableMenuContributor.

2.4.2.1 Ausführliche Beschreibung

Definiert in Zeile 6 der Datei BundleMgrContributor.java.

Öffentliche Methoden

- BundleMgrContributor ()
- String getAdopterId ()
- String getContributionTarget ()
- String getContributorId ()
- String getTargetSubmenu ()

Statische öffentliche Attribute

- static final String BUNDLE_MRG_ADOPTER_ID = "BundleMgrAdopter"
- static final String BUNDLE_MRG_CONTRIBUTOR_ID = "BundleMgrContributor"

2.4.2.2 Beschreibung der Konstruktoren und Destruktoren

BundleMgrContributor ()

Definiert in Zeile 12 der Datei BundleMgrContributor.java.

2.4.2.3 Dokumentation der Elementfunktionen

String getAdopterId ()

Definiert in Zeile 16 der Datei BundleMgrContributor.java.

String getContributionTarget ()

Definiert in Zeile 20 der Datei BundleMgrContributor.java.

String getContributorId ()

Definiert in Zeile 24 der Datei BundleMgrContributor.java.

String getTargetSubmenu ()

Definiert in Zeile 28 der Datei BundleMgrContributor.java.

2.4.2.4 Dokumentation der Datenelemente

final String BUNDLE_MRG_ADOPTER_ID = "BundleMgrAdopter" [static]

Definiert in Zeile 8 der Datei BundleMgrContributor.java.

```
final String BUNDLE_MRG_CONTRIBUTOR_ID = "BundleMgrContributor"  
[static]
```

Definiert in Zeile 10 der Datei BundleMgrContributor.java.

2.4.3 Upload Klassenreferenz

Abgeleitet von javax::servlet::http::HttpServlet.

2.4.3.1 Ausführliche Beschreibung

Autor:

Christian Zillmann Servlet zum Uploaden von Bundles auf den Sever

Definiert in Zeile 25 der Datei Upload.java.

Geschützte Methoden

- void doPost (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

2.4.3.2 Dokumentation der Elementfunktionen

```
void doPost (HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException    [protected]
```

Definiert in Zeile 27 der Datei Upload.java.

3 de.offis.pg.eam.core_usermgr

3.1 BundleConfiguration

3.1.1 BundleViews Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::module::api::IModuleView.

3.1.1.1 Ausführliche Beschreibung

Definiert in Zeile 22 der Datei BundleViews.java.

Öffentliche Methoden

- String getBundleVersion ()
- String getDescription ()
- long getEquinoxBundleId ()
- String getName ()
- List< View > getViews ()

Statische private Attribute

- static final String VIEWS = "bundle_views.xml"

3.1.1.2 Dokumentation der Elementfunktionen

String getBundleVersion ()

Definiert in Zeile 68 der Datei BundleViews.java.

String getDescription ()

Definiert in Zeile 29 der Datei BundleViews.java.

long getEquinoxBundleId ()

Definiert in Zeile 34 der Datei BundleViews.java.

String getName ()

Definiert in Zeile 39 der Datei BundleViews.java.

List<View> getViews ()

Definiert in Zeile 44 der Datei BundleViews.java.

Benutzt BundleViews.VIEWS.

3.1.1.3 Dokumentation der Datenelemente

final String VIEWS = "bundle_views.xml" [static, private]

Definiert in Zeile 25 der Datei BundleViews.java.

Wird benutzt von BundleViews.getViews().

3.1.2 Configuration Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::configuration::api::IBundleConfiguration`.

3.1.2.1 Ausführliche Beschreibung

Definiert in Zeile 10 der Datei Configuration.java.

Öffentliche Methoden

- `boolean bundleHasObjects ()`
- `boolean bundleHasRelations ()`
- `boolean bundleHasViews ()`
- `Document getObjectsJdomDocument ()` throws `JDOMException`, `IOException`

3.1.2.2 Dokumentation der Elementfunktionen

boolean bundleHasObjects ()

Definiert in Zeile 13 der Datei Configuration.java.

boolean bundleHasRelations ()

Definiert in Zeile 28 der Datei Configuration.java.

boolean bundleHasViews ()

Definiert in Zeile 18 der Datei Configuration.java.

Document `getObjectsJdomDocument ()` throws `JDOMException`, `IOException`

Definiert in Zeile 23 der Datei `Configuration.java`.

3.1.3 Activator Klassenreferenz

Abgeleitet von `org::osgi::framework::BundleActivator`.

3.1.3.1 Ausführliche Beschreibung

Definiert in Zeile 41 der Datei `Activator.java`.

Öffentliche Methoden

- `void start (BundleContext context) throws Exception`
- `void stop (BundleContext context) throws Exception`

Statische öffentliche Attribute

- `static BundleContext bundleContext`
- `static String bundleSymbolicName`
- `static String bundleVersion`
- `static long equinoxBundleId`

Private Attribute

- `ServiceTracker httpServiceTracker`

Klassen

- `class HttpServiceTracker`
- `class ServletContextListenerServletAdaptor`

3.1.3.2 Dokumentation der Elementfunktionen

`void start (BundleContext context) throws Exception`

Definiert in Zeile 53 der Datei `Activator.java`.

Benutzt `Activator.bundleContext`, `Activator.bundleSymbolicName`, `Activator.bundleVersion`, `Activator.equinoxBundleId` und `Activator.httpServiceTracker`.

void stop (BundleContext context) throws Exception

Definiert in Zeile 71 der Datei Activator.java.

Benutzt Activator.httpServiceTracker.

3.1.3.3 Dokumentation der Datenelemente

BundleContext bundleContext [static]

Definiert in Zeile 48 der Datei Activator.java.

Wird benutzt von Activator.start().

String bundleSymbolicName [static]

Definiert in Zeile 47 der Datei Activator.java.

Wird benutzt von Activator.start().

String bundleVersion [static]

Definiert in Zeile 49 der Datei Activator.java.

Wird benutzt von Activator.start().

long equinoxBundleId [static]

Definiert in Zeile 46 der Datei Activator.java.

Wird benutzt von Activator.start().

ServiceTracker httpServiceTracker [private]

Definiert in Zeile 43 der Datei Activator.java.

Wird benutzt von Activator.start() und Activator.stop().

3.1.4 Activator.HttpServiceTracker Klassenreferenz

Abgeleitet von org::osgi::util::tracker::ServiceTracker.

3.1.4.1 Ausführliche Beschreibung

Definiert in Zeile 75 der Datei Activator.java.

Öffentliche Methoden

- Object addingService (ServiceReference reference)

- `HttpServiceTracker` (`BundleContext context`)
- `void removedService` (`ServiceReference reference`, `Object service`)

Statische private Attribute

- `static final String PATH = "/userManagement"`
- `static final String WEB_ROOT = "/WebRoot"`

3.1.4.2 Beschreibung der Konstruktoren und Destruktoren

`HttpServiceTracker` (**`BundleContext context`**)

Definiert in Zeile 80 der Datei `Activator.java`.

3.1.4.3 Dokumentation der Elementfunktionen

`Object addingService` (`ServiceReference reference`)

Definiert in Zeile 84 der Datei `Activator.java`.

Benutzt `Activator.HttpServiceTracker.PATH` und `Activator.HttpServiceTracker.WEB_ROOT`.

`void removedService` (`ServiceReference reference`, `Object service`)

Definiert in Zeile 129 der Datei `Activator.java`.

Benutzt `Activator.HttpServiceTracker.PATH`.

3.1.4.4 Dokumentation der Datenelemente

`final String PATH = "/userManagement"` [`static`, `private`]

Definiert in Zeile 77 der Datei `Activator.java`.

Wird benutzt von `Activator.HttpServiceTracker.addingService()` und `Activator.HttpServiceTracker.removedService()`.

`final String WEB_ROOT = "/WebRoot"` [`static`, `private`]

Definiert in Zeile 78 der Datei `Activator.java`.

Wird benutzt von `Activator.HttpServiceTracker.addingService()`.

3.1.5 `Activator.ServletContextListenerServletAdaptor` Klassenreferenz

Abgeleitet von `javax::servlet::Servlet`.

3.1.5.1 Ausführliche Beschreibung

Definiert in Zeile 138 der Datei Activator.java.

Öffentliche Methoden

- void destroy ()
- ServletConfig getServletConfig ()
- String getServletInfo ()
- void init (ServletConfig config) throws ServletException
- void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException
- ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Private Attribute

- ServletConfig config
- Servlet delegate
- ClassLoader jspLoader
- ServletContextListener listener

3.1.5.2 Beschreibung der Konstruktoren und Destruktoren

ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Definiert in Zeile 144 der Datei Activator.java.

3.1.5.3 Dokumentation der Elementfunktionen

void destroy ()

Definiert in Zeile 178 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config, Activator.ServletContextListenerServletAdaptor.delegate, Activator.ServletContextListenerServletAdaptor.jspLoader und Activator.ServletContextListenerServletAdaptor.listener.

ServletConfig getServletConfig ()

Definiert in Zeile 192 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config.

String getServletInfo ()

Definiert in Zeile 196 der Datei Activator.java.

void init (ServletConfig config) throws ServletException

Definiert in Zeile 152 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config, Activator.ServletContextListenerServletAdaptor.delegate, Activator.ServletContextListenerServletAdaptor.jspLoader und Activator.ServletContextListenerServletAdaptor.listener.

void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException

Definiert in Zeile 166 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.delegate und Activator.ServletContextListenerServletAdaptor.jspLoader.

3.1.5.4 Dokumentation der Datenelemente**ServletConfig config [private]**

Definiert in Zeile 139 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(), Activator.ServletContextListenerServletAdaptor.getServletConfig() und Activator.ServletContextListenerServletAdaptor.init().

Servlet delegate [private]

Definiert in Zeile 141 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(), Activator.ServletContextListenerServletAdaptor.init() und Activator.ServletContextListenerServletAdaptor.service().

ClassLoader jspLoader [private]

Definiert in Zeile 142 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(), Activator.ServletContextListenerServletAdaptor.init() und Activator.ServletContextListenerServletAdaptor.service().

ServletContextListener listener [private]

Definiert in Zeile 140 der Datei Activator.java.

Wird benutzt von `Activator.ServletContextListenerServletAdaptor.destroy()` und `Activator.ServletContextListenerServletAdaptor.init()`.

3.1.6 CoreUserMgrContributor Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::menu::api::AbstractAdoptableMenuContributor`.

3.1.6.1 Ausführliche Beschreibung

Definiert in Zeile 6 der Datei `CoreUserMgrContributor.java`.

Öffentliche Methoden

- `CoreUserMgrContributor ()`
- `String getAdopterId ()`
- `String getContributionTarget ()`
- `String getContributorId ()`
- `String getTargetSubmenu ()`

3.1.6.2 Beschreibung der Konstruktoren und Destruktoren

CoreUserMgrContributor ()

Definiert in Zeile 8 der Datei `CoreUserMgrContributor.java`.

3.1.6.3 Dokumentation der Elementfunktionen

String getAdopterId ()

Definiert in Zeile 12 der Datei `CoreUserMgrContributor.java`.

String getContributionTarget ()

Definiert in Zeile 16 der Datei `CoreUserMgrContributor.java`.

String getContributorId ()

Definiert in Zeile 20 der Datei `CoreUserMgrContributor.java`.

String getTargetSubmenu ()

Definiert in Zeile 24 der Datei `CoreUserMgrContributor.java`.

3.2 Managed Beans

3.2.1 GroupOverview Klassenreferenz

3.2.1.1 Ausführliche Beschreibung

Die Bean fuer die Verwaltung von Gruppen. Bearbeitung von Name und Beschreibung
Zuweisung von Benutzern und Rollen

Autor:

Christian R.

Definiert in Zeile 32 der Datei GroupOverview.java.

Öffentliche Methoden

- void addRole () throws NumberFormatException, Throwable
- void addUser () throws NumberFormatException, Throwable
- String back ()
- void deleteAssignedRole () throws NumberFormatException, Throwable
- void deleteAssignedUser () throws NumberFormatException, Throwable
- String deleteGroup () throws NumberFormatException, Throwable
- ArrayList< Role > getAssignedRoles () throws NumberFormatException, Throwable
- ArrayList< User > getAssignedUsers () throws NumberFormatException, Throwable
- Group getGroup ()
- int getGroupId ()
- ArrayList< Group > getGroups () throws NumberFormatException, Throwable
- HtmlInputHidden getHidden ()
- HtmlInputHidden getHidden2 ()
- HtmlInputHidden getHidden3 ()
- ManageMenu getMenu ()
- ArrayList< Role > getNotAssignedRoles () throws NumberFormatException, Throwable
- ArrayList< User > getNotAssignedUsers () throws NumberFormatException, Throwable
- DataModel getResultModel () throws NumberFormatException, Throwable
- String getSortColumnOver ()
- void goAssigned (ActionEvent ae)
- String goToAssignedGroup () throws NumberFormatException, Throwable
- String goToCreateGroup ()
- String goToDeleteFromEdit ()

- String goToDeleteFromOverview ()
- String goToEditAfterCreate () throws NumberFormatException, Throwable
- String goToEditGroup ()
- GroupOverview ()
- boolean isSortAscendingOver ()
- boolean isVisibleLinks ()
- void saveGroup () throws NumberFormatException, Throwable
- void setAssignedRoles (ArrayList< Role > assignedRoles)
- void setAssignedUsers (ArrayList< User > assignedUsers)
- void setGroup (Group group)
- void setGroups (ArrayList< Group > groups)
- void setHidden (HtmlInputHidden hidden)
- void setHidden2 (HtmlInputHidden hidden2)
- void setHidden3 (HtmlInputHidden hidden3)
- void setMenu (ManageMenu menu)
- void setNotAssignedRoles (ArrayList< Role > notAssignedRoles)
- void setNotAssignedUsers (ArrayList< User > notAssignedUsers)
- void setResultModel (DataModel resultModel)
- void setSortAscendingOver (boolean sortAscendingOver)
- void setSortColumnOver (String sortColumnOver)
- void validateName (FacesContext context, UIComponent component, Object value) throws ValidatorException

Private Attribute

- ArrayList< Role > assignedRoles
- ArrayList< User > assignedUsers
- Group group
- int groupId
- ArrayList< Group > groups
- HtmlInputHidden hidden
- HtmlInputHidden hidden2
- HtmlInputHidden hidden3
- ManageMenu menu
- ArrayList< Role > notAssignedRoles
- ArrayList< User > notAssignedUsers
- DataModel resultModel
- boolean sortAscendingOver = true
- String sortColumnOver = null

- boolean visibleLinks

3.2.1.2 Beschreibung der Konstruktoren und Destruktoren

GroupOverview ()

Definiert in Zeile 51 der Datei GroupOverview.java.

Benutzt GroupOverview.assignedRoles, GroupOverview.assignedUsers, GroupOverview.group, GroupOverview.groupId, GroupOverview.groups, GroupOverview.menu, GroupOverview.notAssignedRoles, GroupOverview.notAssignedUsers, GroupOverview.resultModel und GroupOverview.visibleLinks.

3.2.1.3 Dokumentation der Elementfunktionen

void addRole () throws NumberFormatException, Throwable

Fuegt eine Rolle zu einer Gruppe hinzu

Ausnahmebehandlung:

Throwable

NumberFormatException

Definiert in Zeile 341 der Datei GroupOverview.java.

Benutzt GroupOverview.assignedRoles, GroupOverview.groupId und GroupOverview.notAssignedRoles.

void addUser () throws NumberFormatException, Throwable

Fuegt einen Benutzer zu einer Gruppe hinzu

Ausnahmebehandlung:

Throwable

NumberFormatException

Definiert in Zeile 371 der Datei GroupOverview.java.

Benutzt GroupOverview.assignedUsers, GroupOverview.groupId und GroupOverview.notAssignedUsers.

String back ()

Leitet zurueck zur der aufrufenden Instanz. Dies kann zurueck zur Gruppen-Uebersicht sein, aber auch zurueck zu einer aufrufenden Rolle oder einem Benutzer sein.

Rückgabe:

String, für die Navigation mit faces-config

Definiert in Zeile 262 der Datei GroupOverview.java.

Benutzt GroupOverview.assignedRoles, GroupOverview.assignedUsers, GroupOverview.notAssignedRoles, GroupOverview.notAssignedUsers, GroupOverview.sortAscendingOver und GroupOverview.sortColumnOver.

void deleteAssignedRole () throws NumberFormatException, Throwable

Entfernt eine der Gruppe zugewiesene Rolle aus der Zuweisungstabelle

Ausnahmebehandlung:

Throwable

NumberFormatException

Definiert in Zeile 283 der Datei GroupOverview.java.

Benutzt GroupOverview.assignedRoles, GroupOverview.groupId, GroupOverview.hidden2 und GroupOverview.notAssignedRoles.

void deleteAssignedUser () throws NumberFormatException, Throwable

Entfernt einen der Gruppe zugewiesenen Benutzer aus der Zuweisungstabelle

Ausnahmebehandlung:

Throwable

NumberFormatException

Definiert in Zeile 312 der Datei GroupOverview.java.

Benutzt GroupOverview.assignedUsers, GroupOverview.groupId, GroupOverview.hidden3 und GroupOverview.notAssignedUsers.

String deleteGroup () throws NumberFormatException, Throwable

Hier wird eine Gruppe endgueltig aus der Datenbank geloescht.

Rückgabe:

String, für die faces-config

Ausnahmebehandlung:

Throwable

NumberFormatException

Definiert in Zeile 227 der Datei GroupOverview.java.

Benutzt GroupOverview.group, GroupOverview.sortAscendingOver und GroupOverview.sortColumnOver.

ArrayList<Role> getAssignedRoles () throws NumberFormatException, Throwable

Definiert in Zeile 459 der Datei GroupOverview.java.

Benutzt GroupOverview.assignedRoles, GroupOverview.groupId und GroupOverview.setAssignedRoles().

ArrayList<User> getAssignedUsers () throws NumberFormatException, Throwable

Definiert in Zeile 447 der Datei GroupOverview.java.

Benutzt GroupOverview.assignedUsers, GroupOverview.groupId und GroupOverview.setAssignedUsers().

Group getGroup ()

Definiert in Zeile 544 der Datei GroupOverview.java.

Benutzt GroupOverview.group.

int getGroupId ()

Definiert in Zeile 541 der Datei GroupOverview.java.

Benutzt GroupOverview.groupId.

ArrayList<Group> getGroups () throws NumberFormatException, Throwable

Definiert in Zeile 435 der Datei GroupOverview.java.

Benutzt GroupOverview.groups und GroupOverview.setGroups().

Wird benutzt von GroupOverview.getResultModel().

HtmlInputHidden getHidden ()

Definiert in Zeile 520 der Datei GroupOverview.java.

Benutzt GroupOverview.hidden.

HtmlInputHidden getHidden2 ()

Definiert in Zeile 526 der Datei GroupOverview.java.

Benutzt GroupOverview.hidden2.

HtmlInputHidden getHidden3 ()

Definiert in Zeile 532 der Datei GroupOverview.java.

Benutzt GroupOverview.hidden3.

ManageMenu getMenu ()

Definiert in Zeile 550 der Datei GroupOverview.java.

Benutzt GroupOverview.menu.

ArrayList<Role> getNotAssignedRoles () throws NumberFormatException, Throwable

Definiert in Zeile 495 der Datei GroupOverview.java.

Benutzt GroupOverview.assignedRoles und GroupOverview.notAssignedRoles.

ArrayList<User> getNotAssignedUsers () throws NumberFormatException, Throwable

Definiert in Zeile 471 der Datei GroupOverview.java.

Benutzt GroupOverview.assignedUsers und GroupOverview.notAssignedUsers.

DataModel getResultModel () throws NumberFormatException, Throwable

Definiert in Zeile 570 der Datei GroupOverview.java.

Benutzt GroupOverview.getGroups(), GroupOverview.menu, GroupOverview.resultModel, ManageMenu.setSortAscendingAs(), ManageMenu.setSortAscendingAs2(), ManageMenu.setSortAscendingNoAs(), ManageMenu.setSortAscendingNoAs2(), ManageMenu.setSortColumnAs(), ManageMenu.setSortColumnAs2(), ManageMenu.setSortColumnNoAs() und ManageMenu.setSortColumnNoAs2().

String getSortColumnOver ()

Definiert in Zeile 556 der Datei GroupOverview.java.

Benutzt GroupOverview.sortColumnOver.

void goAssigned (ActionEvent ae)

Wird aufgerufen, falls man sich von einer Gruppe aus Details zu einer Rolle oder einem Benutzer ansehen moechte. Hier wird in der Navigationsliste in ManageMenu vermerkt, von wo der Aufruf zu einer anderen Bean kam.

Parameter:

ae,ActionEvent,wird aber nicht genutzt

Definiert in Zeile 120 der Datei GroupOverview.java.

Benutzt GroupOverview.menu, ManageMenu.setSortAscendingAs(), ManageMenu.setSortAscendingAs2(), ManageMenu.setSortAscendingNoAs(), ManageMenu.setSortAscendingNoAs2(), ManageMenu.setSortColumnAs(),

ManageMenu.setSortColumnAs2(), ManageMenu.setSortColumnNoAs() und ManageMenu.setSortColumnNoAs2().

String goToAssignedGroup () throws NumberFormatException, Throwable

Wird aufgerufen, wenn man sich von einer bearbeiteten Rolle oder einem Nutzer aus Details zu einer Gruppe ansehen will.

Rückgabe:

String, für die Navigation mit faces-config

Ausnahmebehandlung:

Throwable

NumberFormatException

Definiert in Zeile 92 der Datei GroupOverview.java.

Benutzt GroupOverview.group, GroupOverview.groupId, GroupOverview.menu, ManageMenu.setSortAscendingAs(), ManageMenu.setSortAscendingAs2(), ManageMenu.setSortAscendingNoAs(), ManageMenu.setSortAscendingNoAs2(), ManageMenu.setSortColumnAs(), ManageMenu.setSortColumnAs2(), ManageMenu.setSortColumnNoAs(), ManageMenu.setSortColumnNoAs2() und GroupOverview.visibleLinks.

String goToCreateGroup ()

Dient zur Weiterleitung nach group_create.jsp.

Rückgabe:

String, für die die faces-config

Definiert in Zeile 160 der Datei GroupOverview.java.

Benutzt GroupOverview.group.

String goToDeleteFromEdit ()

delete-Aufruf von group_edit aus.

Rückgabe:

String, für Navigation mit faces-config

Definiert in Zeile 201 der Datei GroupOverview.java.

String goToDeleteFromOverview ()

delete-Aufruf von group_overview aus.

Rückgabe:

String, für Navigation mit faces-config

Definiert in Zeile 210 der Datei GroupOverview.java.

Benutzt GroupOverview.groupId, GroupOverview.groups, GroupOverview.hidden und GroupOverview.setGroup().

String goToEditAfterCreate () throws NumberFormatException, Throwable

Setzt die notwendigen Werte zum Bearbeiten einer Gruppe. Wird normalerweise nach dem Erstellen einer Gruppe aufgerufen. Unterschied zu goToEditGroup: Die GruppenId muss direkt aus der Datenbank geladen werden.

Rückgabe:

String, für die Navigation mit faces-config

Ausnahmebehandlung:

Throwable

NumberFormatException

Definiert in Zeile 176 der Datei GroupOverview.java.

Benutzt GroupOverview.group, GroupOverview.groupId, GroupOverview.groups und GroupOverview.visibleLinks.

String goToEditGroup ()

Setzt die notwendigen Werte zum Bearbeiten einer Gruppe. Wird normalerweise von der Gruppenuebersicht aus aufgerufen.

Rückgabe:

String, für die Navigation mit faces-config

Definiert in Zeile 70 der Datei GroupOverview.java.

Benutzt GroupOverview.group, GroupOverview.groupId, GroupOverview.groups, GroupOverview.hidden und GroupOverview.visibleLinks.

boolean isSortAscendingOver ()

Definiert in Zeile 562 der Datei GroupOverview.java.

Benutzt GroupOverview.sortAscendingOver.

boolean isVisibleLinks ()

Definiert in Zeile 538 der Datei GroupOverview.java.

Benutzt GroupOverview.visibleLinks.

void saveGroup () throws NumberFormatException, Throwable

Uebertraegt die editierten Werte der Gruppenbezeichnung und -beschreibung in die Datenbank.

Ausnahmebehandlung:

Throwable

NumberFormatException

Definiert in Zeile 139 der Datei GroupOverview.java.

Benutzt GroupOverview.group und GroupOverview.groupId.

void setAssignedRoles (ArrayList< Role > assignedRoles)

Definiert in Zeile 468 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.getAssignedRoles().

void setAssignedUsers (ArrayList< User > assignedUsers)

Definiert in Zeile 456 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.getAssignedUsers().

void setGroup (Group group)

Definiert in Zeile 547 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.goToDeleteFromOverview().

void setGroups (ArrayList< Group > groups)

Definiert in Zeile 444 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.getGroups().

void setHidden (HtmlInputHidden hidden)

Definiert in Zeile 523 der Datei GroupOverview.java.

void setHidden2 (HtmlInputHidden hidden2)

Definiert in Zeile 529 der Datei GroupOverview.java.

void setHidden3 (HtmlInputHidden hidden3)

Definiert in Zeile 535 der Datei GroupOverview.java.

void setMenu (ManageMenu menu)

Definiert in Zeile 553 der Datei GroupOverview.java.

void setNotAssignedRoles (ArrayList< Role > notAssignedRoles)

Definiert in Zeile 517 der Datei GroupOverview.java.

void setNotAssignedUsers (ArrayList< User > notAssignedUsers)

Definiert in Zeile 492 der Datei GroupOverview.java.

void setResultModel (DataModel resultModel)

Definiert in Zeile 588 der Datei GroupOverview.java.

void setSortAscendingOver (boolean sortAscendingOver)

Definiert in Zeile 565 der Datei GroupOverview.java.

Wird benutzt von ManageMenu.goToGroups().

void setSortColumnOver (String sortColumnOver)

Definiert in Zeile 559 der Datei GroupOverview.java.

Wird benutzt von ManageMenu.goToGroups().

**void validateName (FacesContext context, UIComponent component, Object value)
throws ValidatorException**

Fall 1: Falls eine Gruppe schon existiert, die Bezeichnung aber nur geaendert werden soll, wird ueberprueft, ob der neue Name schon vergeben ist. Fall 2: Falls eine Gruppe erstellt werden soll, wird nur ueberprueft, ob der Name schon existiert.

Parameter:

context, FacesContext

component, UIComponent

value, Object, das den Wert des Namensfeldes enthaelt

Ausnahmebehandlung:

ValidatorException, wird geworfen, falls der Name schon vergeben ist.

Definiert in Zeile 408 der Datei GroupOverview.java.

Benutzt GroupOverview.group und GroupOverview.groups.

3.2.1.4 Dokumentation der Datenelemente

ArrayList<Role> assignedRoles [private]

Definiert in Zeile 37 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.addRole(), GroupOverview.back(), GroupOverview.deleteAssignedRole(), GroupOverview.getAssignedRoles(), GroupOverview.getNotAssignedRoles() und GroupOverview.GroupOverview().

ArrayList<User> assignedUsers [private]

Definiert in Zeile 36 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.addUser(), GroupOverview.back(), GroupOverview.deleteAssignedUser(), GroupOverview.getAssignedUsers(), GroupOverview.getNotAssignedUsers() und GroupOverview.GroupOverview().

Group group [private]

Definiert in Zeile 44 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.deleteGroup(), GroupOverview.getGroup(), GroupOverview.goToAssignedGroup(), GroupOverview.goToCreateGroup(), GroupOverview.goToEditAfterCreate(), GroupOverview.goToEditGroup(), GroupOverview.GroupOverview(), GroupOverview.saveGroup() und GroupOverview.validateName().

int groupId [private]

Definiert in Zeile 43 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.addRole(), GroupOverview.addUser(), GroupOverview.deleteAssignedRole(), GroupOverview.deleteAssignedUser(), GroupOverview.getAssignedRoles(), GroupOverview.getAssignedUsers(), GroupOverview.getGroupId(), GroupOverview.goToAssignedGroup(), GroupOverview.goToDeleteFromOverview(), GroupOverview.goToEditAfterCreate(), GroupOverview.goToEditGroup(), GroupOverview.GroupOverview() und GroupOverview.saveGroup().

ArrayList<Group> groups [private]

Definiert in Zeile 35 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.getGroups(), GroupOverview.goToDeleteFromOverview(), GroupOverview.goToEditAfterCreate(), GroupOverview.goToEditGroup(), GroupOverview.GroupOverview() und GroupOverview.validateName().

HtmlInputHidden hidden [private]

Definiert in Zeile 40 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.getHidden(),
GroupOverview.goToDeleteFromOverview() und GroupOverview.goToEditGroup().

HtmlInputHidden hidden2 [private]

Definiert in Zeile 41 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.deleteAssignedRole() und
GroupOverview.getHidden2().

HtmlInputHidden hidden3 [private]

Definiert in Zeile 42 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.deleteAssignedUser() und
GroupOverview.getHidden3().

ManageMenu menu [private]

Definiert in Zeile 45 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.getMenu(), GroupOverview.getResultModel(),
GroupOverview.goAssigned(), GroupOverview.goToAssignedGroup() und
GroupOverview.GroupOverview().

ArrayList<Role> notAssignedRoles [private]

Definiert in Zeile 39 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.addRole(), GroupOverview.back(),
GroupOverview.deleteAssignedRole(), GroupOverview.getNotAssignedRoles() und
GroupOverview.GroupOverview().

ArrayList<User> notAssignedUsers [private]

Definiert in Zeile 38 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.addUser(), GroupOverview.back(),
GroupOverview.deleteAssignedUser(), GroupOverview.getNotAssignedUsers() und
GroupOverview.GroupOverview().

DataModel resultModel [private]

Definiert in Zeile 48 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.getResultModel() und
GroupOverview.GroupOverview().

boolean sortAscendingOver = true [private]

Definiert in Zeile 47 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.back(), GroupOverview.deleteGroup() und GroupOverview.isSortAscendingOver().

String sortColumnOver = null [private]

Definiert in Zeile 46 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.back(), GroupOverview.deleteGroup() und GroupOverview.getSortColumnOver().

boolean visibleLinks [private]

Definiert in Zeile 33 der Datei GroupOverview.java.

Wird benutzt von GroupOverview.goToAssignedGroup(), GroupOverview.goToEditAfterCreate(), GroupOverview.goToEditGroup(), GroupOverview.GroupOverview() und GroupOverview.isVisibleLinks().

3.2.2 ManageMenu Klassenreferenz

3.2.2.1 Ausführliche Beschreibung

Unterstützt bei der Navigation zwischen den Beans.

Autor:

Christian R.

Definiert in Zeile 14 der Datei ManageMenu.java.

Öffentliche Methoden

- String getSortColumnAs ()
- String getSortColumnAs2 ()
- String getSortColumnNoAs ()
- String getSortColumnNoAs2 ()
- String goToGroups ()
- String goToRoles ()
- String goToUsers ()
- boolean isSortAscendingAs ()
- boolean isSortAscendingAs2 ()
- boolean isSortAscendingNoAs ()
- boolean isSortAscendingNoAs2 ()

- `ManageMenu ()`
- `void setSortAscendingAs (boolean sortAscendingAs)`
- `void setSortAscendingAs2 (boolean sortAscendingAs2)`
- `void setSortAscendingNoAs (boolean sortAscendingNoAs)`
- `void setSortAscendingNoAs2 (boolean sortAscendingNoAs2)`
- `void setSortColumnAs (String sortColumnAs)`
- `void setSortColumnAs2 (String sortColumnAs2)`
- `void setSortColumnNoAs (String sortColumnNoAs)`
- `void setSortColumnNoAs2 (String sortColumnNoAs2)`

Öffentliche, statische Methoden

- `static void addToFromList (String view)`
- `static String back ()`
- `static void clearFrom ()`
- `static ArrayList< String > getFrom ()`

Statische öffentliche Attribute

- `static ArrayList< String > from = new ArrayList<String>()`
- `static final String groupConfirmDelete = "groupConfirmDelete"`
- `static final String groupCreate = "groupCreate"`
- `static final String groupCreated = "groupCreated"`
- `static final String groupEdit = "groupEdit"`
- `static final String groupOverview = "groupOver"`
- `static final String roleConfirmDelete = "roleConfirmDelete"`
- `static final String roleCreate = "roleCreate"`
- `static final String roleCreated = "roleCreated"`
- `static final String roleEdit = "roleEdit"`
- `static final String roleOverview = "roleOver"`
- `static final String userConfirmDelete = "userConfirmDelete"`
- `static final String userCreate = "userCreate"`
- `static final String userCreated = "userCreated"`
- `static final String userDelete = "userDelete"`
- `static final String userEdit = "userEdit"`
- `static final String userOverview = "userOver"`

Private Attribute

- boolean sortAscendingAs = true
- boolean sortAscendingAs2 = true
- boolean sortAscendingNoAs = true
- boolean sortAscendingNoAs2 = true
- String sortColumnAs = null
- String sortColumnAs2 = null
- String sortColumnNoAs = null
- String sortColumnNoAs2 = null

3.2.2.2 Beschreibung der Konstruktoren und Destruktoren**ManageMenu ()**

Definiert in Zeile 53 der Datei ManageMenu.java.

3.2.2.3 Dokumentation der Elementfunktionen**static void addToFromList (String view) [static]**

Fuegt der Navigationsliste einen neuen Wert hinzu

Parameter:**view**

Definiert in Zeile 75 der Datei ManageMenu.java.

Benutzt ManageMenu.from.

static String back () [static]

Liefert fuer die Navigation innerhalb der einzelnen Sichten die Werte fuer ein "zurueck"

Rückgabe:

String, den entsprechenden ZurueckWert

Definiert in Zeile 65 der Datei ManageMenu.java.

Benutzt ManageMenu.from.

static void clearFrom () [static]

Entfernt alle Elemente aus der NavigationsListe

Definiert in Zeile 82 der Datei ManageMenu.java.

Benutzt ManageMenu.from.

static ArrayList<String> getFrom () [static]

Definiert in Zeile 86 der Datei ManageMenu.java.

Benutzt ManageMenu.from.

String getSortColumnAs ()

Definiert in Zeile 142 der Datei ManageMenu.java.

Benutzt ManageMenu.sortColumnAs.

String getSortColumnAs2 ()

Definiert in Zeile 166 der Datei ManageMenu.java.

Benutzt ManageMenu.sortColumnAs2.

String getSortColumnNoAs ()

Definiert in Zeile 154 der Datei ManageMenu.java.

Benutzt ManageMenu.sortColumnNoAs.

String getSortColumnNoAs2 ()

Definiert in Zeile 178 der Datei ManageMenu.java.

Benutzt ManageMenu.sortColumnNoAs2.

String goToGroups ()

Leitet weiter zur Gruppenuebersicht. Holt das groupOverview-Objekt der Sesssion und setzt in diesem die Werte für die Sortierung der Tabelle.

Rückgabe:

groupOverview, String, der in der faces-config dafür sorgt, zur Gruppenuebersicht weitergeleitet zu werden.

Definiert in Zeile 114 der Datei ManageMenu.java.

Benutzt ManageMenu.groupOverview, GroupOverview.setSortAscendingOver() und GroupOverview.setSortColumnOver().

String goToRoles ()

Leitet weiter zur Rollenuebersicht. Holt das roleOverview-Objekt der Sesssion und setzt in diesem die Werte für die Sortierung der Tabelle.

Rückgabe:

roleOverview, String, der in der faces-config dafür sorgt, zur Rollenuübersicht weitergeleitet zu werden.

Definiert in Zeile 97 der Datei ManageMenu.java.

Benutzt ManageMenu.roleOverview, RoleOverview.setSortAscendingOver() und RoleOverview.setSortColumnOver().

String goToUsers ()

Leitet weiter zur Benutzerübersicht. Holt das userOverview-Objekt der Session und setzt in diesem die Werte für die Sortierung der Tabelle.

Rückgabe:

userOverview, String, der in der faces-config dafür sorgt, zur Benutzerübersicht weitergeleitet zu werden.

Definiert in Zeile 131 der Datei ManageMenu.java.

Benutzt UserOverview.setSortAscendingOver(), UserOverview.setSortColumnOver() und ManageMenu.userOverview.

boolean isSortAscendingAs ()

Definiert in Zeile 148 der Datei ManageMenu.java.

Benutzt ManageMenu.sortAscendingAs.

boolean isSortAscendingAs2 ()

Definiert in Zeile 172 der Datei ManageMenu.java.

Benutzt ManageMenu.sortAscendingAs2.

boolean isSortAscendingNoAs ()

Definiert in Zeile 160 der Datei ManageMenu.java.

Benutzt ManageMenu.sortAscendingNoAs.

boolean isSortAscendingNoAs2 ()

Definiert in Zeile 184 der Datei ManageMenu.java.

Benutzt ManageMenu.sortAscendingNoAs2.

void setSortAscendingAs (boolean sortAscendingAs)

Definiert in Zeile 151 der Datei ManageMenu.java.

Wird benutzt von RoleOverview.editAssignedRole(), UserOverview.editAssignedUser(), GroupOverview.getResultModel(), RoleOverview.getRoles(), RoleOverview.goAssigned(), GroupOverview.goAssigned(), GroupOverview.goToAssignedGroup() und RoleOverview.goToEditRole().

void setSortAscendingAs2 (boolean sortAscendingAs2)

Definiert in Zeile 175 der Datei ManageMenu.java.

Wird benutzt von RoleOverview.editAssignedRole(), UserOverview.editAssignedUser(), GroupOverview.getResultModel(), RoleOverview.getRoles(), RoleOverview.goAssigned(), GroupOverview.goAssigned(), GroupOverview.goToAssignedGroup() und RoleOverview.goToEditRole().

void setSortAscendingNoAs (boolean sortAscendingNoAs)

Definiert in Zeile 163 der Datei ManageMenu.java.

Wird benutzt von RoleOverview.editAssignedRole(), UserOverview.editAssignedUser(), GroupOverview.getResultModel(), RoleOverview.getRoles(), RoleOverview.goAssigned(), GroupOverview.goAssigned(), GroupOverview.goToAssignedGroup() und RoleOverview.goToEditRole().

void setSortAscendingNoAs2 (boolean sortAscendingNoAs2)

Definiert in Zeile 187 der Datei ManageMenu.java.

Wird benutzt von RoleOverview.editAssignedRole(), UserOverview.editAssignedUser(), GroupOverview.getResultModel(), RoleOverview.getRoles(), RoleOverview.goAssigned(), GroupOverview.goAssigned(), GroupOverview.goToAssignedGroup() und RoleOverview.goToEditRole().

void setSortColumnAs (String sortColumnAs)

Definiert in Zeile 145 der Datei ManageMenu.java.

Wird benutzt von RoleOverview.editAssignedRole(), UserOverview.editAssignedUser(), GroupOverview.getResultModel(), RoleOverview.getRoles(), RoleOverview.goAssigned(), GroupOverview.goAssigned(), GroupOverview.goToAssignedGroup() und RoleOverview.goToEditRole().

void setSortColumnAs2 (String sortColumnAs2)

Definiert in Zeile 169 der Datei ManageMenu.java.

Wird benutzt von RoleOverview.editAssignedRole(), UserOverview.editAssignedUser(), GroupOverview.getResultModel(), RoleOverview.getRoles(), RoleOverview.goAssigned(), GroupOverview.goAssigned(), GroupOverview.goToAssignedGroup() und RoleOverview.goToEditRole().

void setSortColumnNoAs (String sortColumnNoAs)

Definiert in Zeile 157 der Datei ManageMenu.java.

Wird benutzt von RoleOverview.editAssignedRole(), UserOverview.editAssignedUser(), GroupOverview.getResultModel(), RoleOverview.getRoles(), RoleOverview.goAssigned(), GroupOverview.goAssigned(), GroupOverview.goToAssignedGroup() und RoleOverview.goToEditRole().

void setSortColumnNoAs2 (String sortColumnNoAs2)

Definiert in Zeile 181 der Datei ManageMenu.java.

Wird benutzt von RoleOverview.editAssignedRole(), UserOverview.editAssignedUser(), GroupOverview.getResultModel(), RoleOverview.getRoles(), RoleOverview.goAssigned(), GroupOverview.goAssigned(), GroupOverview.goToAssignedGroup() und RoleOverview.goToEditRole().

3.2.2.4 Dokumentation der Datenelemente**ArrayList<String> from = new ArrayList<String>() [static]**

Definiert in Zeile 36 der Datei ManageMenu.java.

Wird benutzt von ManageMenu.addToFromList(), ManageMenu.back(), ManageMenu.clearFrom() und ManageMenu.getFrom().

final String groupConfirmDelete = "groupConfirmDelete" [static]

Definiert in Zeile 33 der Datei ManageMenu.java.

final String groupCreate = "groupCreate" [static]

Definiert in Zeile 32 der Datei ManageMenu.java.

final String groupCreated = "groupCreated" [static]

Definiert in Zeile 31 der Datei ManageMenu.java.

final String groupEdit = "groupEdit" [static]

Definiert in Zeile 30 der Datei ManageMenu.java.

final String groupOverview = "groupOver" [static]

Definiert in Zeile 29 der Datei ManageMenu.java.

Wird benutzt von ManageMenu.goToGroups().

final String roleConfirmDelete = "roleConfirmDelete" [static]

Definiert in Zeile 22 der Datei ManageMenu.java.

final String roleCreate = "roleCreate" [static]

Definiert in Zeile 20 der Datei ManageMenu.java.

final String roleCreated = "roleCreated" [static]

Definiert in Zeile 21 der Datei ManageMenu.java.

final String roleEdit = "roleEdit" [static]

Definiert in Zeile 19 der Datei ManageMenu.java.

final String roleOverview = "roleOver" [static]

Definiert in Zeile 18 der Datei ManageMenu.java.

Wird benutzt von ManageMenu.goToRoles().

boolean sortAscendingAs = true [private]

Definiert in Zeile 40 der Datei ManageMenu.java.

Wird benutzt von ManageMenu.isSortAscendingAs().

boolean sortAscendingAs2 = true [private]

Definiert in Zeile 45 der Datei ManageMenu.java.

Wird benutzt von ManageMenu.isSortAscendingAs2().

boolean sortAscendingNoAs = true [private]

Definiert in Zeile 42 der Datei ManageMenu.java.

Wird benutzt von ManageMenu.isSortAscendingNoAs().

boolean sortAscendingNoAs2 = true [private]

Definiert in Zeile 47 der Datei ManageMenu.java.

Wird benutzt von ManageMenu.isSortAscendingNoAs2().

String sortColumnAs = null [private]

Definiert in Zeile 39 der Datei ManageMenu.java.

Wird benutzt von ManageMenu.getSortColumnAs().

String sortColumnAs2 = null [private]

Definiert in Zeile 44 der Datei ManageMenu.java.

Wird benutzt von ManageMenu.getSortColumnAs2().

String sortColumnNoAs = null [private]

Definiert in Zeile 41 der Datei ManageMenu.java.

Wird benutzt von ManageMenu.getSortColumnNoAs().

String sortColumnNoAs2 = null [private]

Definiert in Zeile 46 der Datei ManageMenu.java.

Wird benutzt von ManageMenu.getSortColumnNoAs2().

final String userConfirmDelete = "userConfirmDelete" [static]

Definiert in Zeile 28 der Datei ManageMenu.java.

final String userCreate = "userCreate" [static]

Definiert in Zeile 25 der Datei ManageMenu.java.

final String userCreated = "userCreated" [static]

Definiert in Zeile 26 der Datei ManageMenu.java.

final String userDelete = "userDelete" [static]

Definiert in Zeile 27 der Datei ManageMenu.java.

final String userEdit = "userEdit" [static]

Definiert in Zeile 24 der Datei ManageMenu.java.

final String userOverview = "userOver" [static]

Definiert in Zeile 23 der Datei ManageMenu.java.

Wird benutzt von ManageMenu.goToUsers().

3.2.3 RoleOverview Klassenreferenz

3.2.3.1 Ausführliche Beschreibung

Bean fuer die Verwaltung von Rollen. Speziell: Bearbeitung von Name und Beschreibung
Benutzer- und Gruppenzuweisung Zuweisung von Instanzrechten Zuweisung von
Metamodellrechten Zuweisung von Sichten und damit Rechten ueber Methoden Zuweisung
von Queries

Autor:

Christian R., Igor S.

Definiert in Zeile 52 der Datei RoleOverview.java.

Öffentliche Methoden

- void addGroup () throws Throwable
- void addUser () throws Throwable
- String back ()
- int convertAccess ()
- String createRole () throws Throwable
- void deleteGroup () throws Throwable
- String deleteRight () throws Throwable
- String deleteRole () throws Throwable
- void deleteSelectedFunctions () throws Throwable
- String deleteSelectedRights () throws Throwable
- void deleteUser () throws Throwable
- String editAssignedRole () throws Throwable
- void editRole () throws Throwable
- ArrayList< AuthInfo > getAssignedFilteredRights ()
- LinkedList< Query > getAssignedQueries () throws Throwable
- ArrayList< AuthInfo > getAssignedRights () throws Throwable
- ArrayList< AuthInfo > getAttr ()
- AuthInfo getAttrInfoById (int id)
- ArrayList< AuthInfo > getAttrObj ()
- ArrayList< AuthInfo > getAttrRel ()
- ArrayList< AuthInfo > getAttrTypes ()
- ArrayList< SelectItem > getBundleList ()
- String getBundleOneList ()
- String getBundleOneMenu ()
- ArrayList< Module > getBundles ()

- `int getColumnNumbers ()`
- `String[] getCurrentAttr ()`
- `String[] getCurrentAttrObj ()`
- `String[] getCurrentAttrRel ()`
- `String[] getCurrentAttrTypes ()`
- `String[] getCurrentMet ()`
- `String[] getCurrentObj ()`
- `String[] getCurrentObjByType ()`
- `String[] getCurrentObjTypes ()`
- `String[] getCurrentRel ()`
- `String[] getCurrentRelTypes ()`
- `String[] getDefaultAccess ()`
- `ArrayList< View > getFiltAssignedRights ()`
- `ArrayList< Group > getGroups ()` throws `Throwable`
- `ArrayList< Group > getGroupsByRole ()` throws `Throwable`
- `HtmlInputHidden getHidden ()`
- `HtmlInputHidden getHidden2 ()`
- `HtmlInputHidden getHidden3 ()`
- `HtmlInputHidden getHiddenRoleId ()`
- `HtmlInputHidden getHiddenView ()`
- `ManageMenu getMenu ()`
- `ArrayList< AuthInfo > getMeta ()`
- `AuthInfo getMetInfoById (int id)`
- `ArrayList< Group > getNotAssignedGroups ()` throws `Throwable`
- `LinkedList< Query > getNotAssignedQueries ()` throws `Throwable`
- `ArrayList< User > getNotAssignedUsers ()` throws `Throwable`
- `AuthInfo getObjAttributeInfoById (int id)`
- `ArrayList< AuthInfo > getObjByType ()`
- `AuthInfo getObjByTypeInfoById (int id)`
- `ArrayList< AuthInfo > getObjects ()`
- `AuthInfo getObjInfoById (int id)`
- `AuthInfo getObjTypeInfoById (int id)`
- `ArrayList< AuthInfo > getObjTypes ()`
- `ArrayList< AuthInfo > getRel ()`
- `AuthInfo getRelationById (int id)`
- `ArrayList< AuthInfo > getRelations ()`
- `AuthInfo getRelAttributeInfoById (int id)`
- `AuthInfo getRelInfoById (int id)`

- AuthInfo getRelTypeInfoById (int id)
- ArrayList< AuthInfo > getRelTypes ()
- String getRightTypeChoice ()
- Role getRole ()
- int getRoleId ()
- ArrayList< Role > getRoles () throws Throwable
- ArrayList< SelectItem > getSelAccess () throws Throwable
- ArrayList< SelectItem > getSelAttr () throws Throwable
- ArrayList< SelectItem > getSelAttrObj () throws Throwable
- ArrayList< SelectItem > getSelAttrRel () throws Throwable
- ArrayList< SelectItem > getSelectableRights ()
- ArrayList< String > getSelectedRights ()
- ArrayList< SelectItem > getSelMet () throws Throwable
- ArrayList< SelectItem > getSelObj () throws Throwable
- ArrayList< SelectItem > getSelObjByType () throws Throwable
- ArrayList< SelectItem > getSelObjType () throws Throwable
- ArrayList< SelectItem > getSelRel () throws Throwable
- ArrayList< SelectItem > getSelRelTypes () throws Throwable
- String getSortColumnOver ()
- String getSortQueriesAssignedColumn ()
- String getSortQueriesNotAssignedColumn ()
- String getStandardOneMenu ()
- ArrayList< SelectItem > getStandardRoles ()
- String getTableName ()
- ArrayList< User > getUsers () throws Throwable
- ArrayList< User > getUsersByRole () throws Throwable
- void goAssigned (ActionEvent ae)
- String goToCreateNewRole ()
- String goToDeleteFromEdit ()
- String goToDeleteFromOverview ()
- String goToEditRole () throws Throwable
- boolean isBundleSelected ()
- boolean isCreateAttrTyp ()
- boolean isCreateMeta ()
- boolean isCreateObjType ()
- boolean isCreateRelType ()
- boolean isCreateStatus ()
- boolean isCreateTime ()

- boolean isInAssignedRights (String value, String object)
- boolean isMetaTypeRight ()
- boolean isSortAscendingOver ()
- boolean isSortQueriesAssignedAscending ()
- boolean isSortQueriesNotAssignedAscending ()
- boolean isStandardSelected ()
- boolean isVisibleLinks ()
- void listenerCreateAttrTyp (ValueChangeEvent event) throws ProxyException, Throwable
- void listenerCreateMeta (ValueChangeEvent event) throws ProxyException, Throwable
- void listenerCreateObjType (ValueChangeEvent event) throws ProxyException, Throwable
- void listenerCreateRelType (ValueChangeEvent event) throws ProxyException, Throwable
- void listenerCreateStatus (ValueChangeEvent event) throws ProxyException, Throwable
- void listenerCreateTime (ValueChangeEvent event) throws ProxyException, Throwable
- void listenerSelBundle (ValueChangeEvent event)
- void listenerSelStandard (ValueChangeEvent event)
- void loadFuncRights () throws Throwable
- void modelSelected (ValueChangeEvent event)
- void objByTypeSelected (ValueChangeEvent event)
- void objSelected (ValueChangeEvent event)
- void objTypeSelected (ValueChangeEvent event)
- void queryAssign () throws Throwable
- void queryRemoveOne () throws Throwable
- void relByTypeSelected (ValueChangeEvent event)
- void relTypeSelected (ValueChangeEvent event)
- void removeOneAssignment () throws Throwable
- void resolveMetamodelDependencies (int modelId, int access) throws Throwable
- void resolveObjByTypeDependencies (int access) throws Throwable
- void resolveObjectDependencies (int access) throws Throwable
- void resolveObjecttypeDependencies (int access) throws Throwable
- void resolveRelationtypeDependencies (int access) throws Throwable
- void resolveRelByTypeDependencies (int access) throws Throwable
- RoleOverview ()
- String saveAccess () throws Throwable

- void saveAssignments () throws Throwable
- void saveRights () throws Throwable
- String saveRightsByObjType () throws Throwable
- String saveRightsByRelType () throws Throwable
- void setAttr (ArrayList< AuthInfo > attr)
- void setAttrObj (ArrayList< AuthInfo > attrObj)
- void setAttrRel (ArrayList< AuthInfo > attrRel)
- void setAttrTypes (ArrayList< AuthInfo > attrTypes)
- void setBundleList (ArrayList< SelectItem > bundleList)
- void setBundleOneList (String bundleOneList)
- void setBundleOneMenu (String bundleOneMenu)
- void setBundles (ArrayList< Module > bundles)
- void setBundleSelected (boolean bundleSelected)
- void setColumnNumbers (int columnNumbers)
- void setCreateAttrTyp (boolean createAttrTyp)
- void setCreateMeta (boolean createMeta)
- void setCreateObjType (boolean createObjType)
- void setCreateRelType (boolean createRelType)
- void setCreateStatus (boolean createTime)
- void setCreateTime (boolean createTime)
- void setCurrentAttr (String[] currentAttr)
- void setCurrentAttrObj (String[] currentAttrObj)
- void setCurrentAttrRel (String[] currentAttrRel)
- void setCurrentAttrTypes (String[] currentAttrTypes)
- void setCurrentMet (String[] currentMet)
- void setCurrentObj (String[] currentObj)
- void setCurrentObjByType (String[] currentObjByType)
- void setCurrentObjTypes (String[] currentObjTypes)
- void setCurrentRel (String[] currentRel)
- void setCurrentRelTypes (String[] currentRelTypes)
- void setDefaultAccess (String[] defaultAccess)
- void setFiltAssignedRights (ArrayList< View > filtAssignedRights)
- void setGroups (ArrayList< Group > groups)
- void setHidden (HtmlInputHidden hidden)
- void setHidden2 (HtmlInputHidden hidden2)
- void setHidden3 (HtmlInputHidden hidden3)
- void setHiddenRoleId (HtmlInputHidden hiddenRoleId)
- void setHiddenView (HtmlInputHidden hiddenView)

- void setMenu (ManageMenu menu)
- void setMetaTypeRight (boolean metaTypeRight)
- void setNotAssignedGroups (ArrayList< Group > notAssignedGroups)
- void setObjByType (ArrayList< AuthInfo > objByType)
- void setObjects (ArrayList< AuthInfo > objects)
- void setObjTypes (ArrayList< AuthInfo > objTypes)
- void setRel (ArrayList< AuthInfo > rel)
- void setRelations (ArrayList< AuthInfo > relations)
- void setRelTypes (ArrayList< AuthInfo > relTypes)
- void setRightTypeChoice (String rightTypeChoice)
- void setRoleId (int roleId)
- void setRoles (ArrayList< Role > roles)
- void setSelAccess (ArrayList< SelectItem > selAccess)
- void setSelAttr (ArrayList< SelectItem > selAttr)
- void setSelAttrObj (ArrayList< SelectItem > selAttrObj)
- void setSelAttrRel (ArrayList< SelectItem > selAttrRel)
- void setSelectableRights (ArrayList< SelectItem > selectableRights)
- void setSelectedRights (ArrayList< String > selectedBundleRoles)
- void setSelMet (ArrayList< SelectItem > selMet)
- void setSelObj (ArrayList< SelectItem > selObj)
- void setSelObjByType (ArrayList< SelectItem > selObjByType)
- void setSelObjType (ArrayList< SelectItem > selObjType)
- void setSelRel (ArrayList< SelectItem > selRel)
- void setSelRelTypes (ArrayList< SelectItem > selRelTypes)
- void setSortAscendingOver (boolean sortAscendingOver)
- void setSortColumnOver (String sortColumnOver)
- void setSortQueriesAssignedAscending (boolean sortQueriesAssignedAscending)
- void setSortQueriesAssignedColumn (String sortQueriesAssignedColumn)
- void setSortQueriesNotAssignedAscending (boolean sortQueriesNotAssignedAscending)
- void setSortQueriesNotAssignedColumn (String sortQueriesNotAssignedColumn)
- void setStandardOneMenu (String standardOneMenu)
- void setStandardRoles (ArrayList< SelectItem > standardRoles)
- void setStandardSelected (boolean standardSelected)
- void setTableName (String tableName)
- void setUsers (ArrayList< User > users)
- void switchDataMeta (ValueChangeEvent event)
- void validateName (FacesContext context, UIComponent component, Object value)

throws `ValidatorException`

Paketattribute

- `String rightTypeChoice = "true"`

Private Methoden

- `boolean alreadyAssigned (long rightId)`
- `IDalRole generateDalRoleObject ()` throws `ProxyException`, `Throwable`
- `void setCreateAccessDefault ()`

Private Attribute

- `ArrayList< AuthInfo > assignedFilteredRights`
- `ArrayList< View > assignedFunc = new ArrayList<View>()`
- `LinkedList< Query > assignedQueries`
- `ArrayList< AuthInfo > assignedRights`
- `ArrayList< AuthInfo > attr`
- `final String ATTRIBUTE_RIGHT = IAuthNames.METAMODEL_ATTRIBUTE`
- `ArrayList< AuthInfo > attrObj`
- `ArrayList< AuthInfo > attrRel`
- `ArrayList< AuthInfo > attrTypes`
- `String bundleId = "0"`
- `ArrayList< SelectItem > bundleList = new ArrayList<SelectItem>()`
- `String bundleOneList = ""`
- `String bundleOneMenu = ""`
- `ArrayList< Module > bundles = new ArrayList<Module>()`
- `boolean bundleSelected = false`
- `int columnNumbers = 2`
- `boolean[] createRight = {false, false, false, false, false, false}`
- `String[] currentAttr`
- `String[] currentAttrObj`
- `String[] currentAttrRel`
- `String[] currentAttrTypes`
- `String[] currentMet`
- `String[] currentObj`
- `String[] currentObjByType`
- `String[] currentObjTypes`

- String[] currentRel
- String[] currentRelTypes
- String[] defaultAccess
- ArrayList< View > filtAssignedRights = new ArrayList<View>()
- ArrayList< Group > groups
- HtmlInputHidden hidden
- HtmlInputHidden hidden2
- HtmlInputHidden hidden3
- HtmlInputHidden hiddenRoleId
- HtmlInputHidden hiddenView
- boolean listLoaded = false
- ManageMenu menu
- ArrayList< AuthInfo > meta
- final String METAMODEL_RIGHT = IAuthNames.METAMODEL
- boolean metaSelected = false
- boolean metaTypeRight = true
- final String noFilter = "nofilter"
- ArrayList< Group > notAssignedGroups
- LinkedList< Query > notAssignedQueries
- ArrayList< User > notAssignedUsers
- ArrayList< AuthInfo > objByType
- boolean objByTypeSelected = false
- final String OBJECT_RIGHT = IAuthNames.METAMODEL_OBJECT
- final String OBJECT_TYPE_RIGHT =
IAuthNames.METAMODEL_OBJECT_TYPE
- ArrayList< AuthInfo > objects
- boolean objSelected = false
- ArrayList< AuthInfo > objTypes
- boolean objTypeSelected = false
- ArrayList< AuthInfo > rel
- final String RELATION_RIGHT = IAuthNames.METAMODEL_RELATION
- final String RELATION_TYPE_RIGHT =
IAuthNames.METAMODEL_RELATION_TYPE
- ArrayList< AuthInfo > relations
- boolean relByTypeSelected = false
- ArrayList< AuthInfo > relTypes
- boolean relTypeSelected = false
- Role role

- int roleId
- ArrayList< Role > roles
- ArrayList< SelectItem > selAccess
- ArrayList< SelectItem > selAttr
- ArrayList< SelectItem > selAttrObj
- ArrayList< SelectItem > selAttrRel
- ArrayList< SelectItem > selectableRights = new ArrayList<SelectItem>()
- boolean selectAllRights = false
- ArrayList< String > selectedRights = new ArrayList<String>()
- ArrayList< SelectItem > selMet
- ArrayList< SelectItem > selObj
- ArrayList< SelectItem > selObjByType
- ArrayList< SelectItem > selObjType
- ArrayList< SelectItem > selRel
- ArrayList< SelectItem > selRelTypes
- boolean sortAscendingOver = true
- String sortColumnOver = null
- boolean sortQueriesAssignedAscending = true
- String sortQueriesAssignedColumn = null
- boolean sortQueriesNotAssignedAscending = true
- String sortQueriesNotAssignedColumn = null
- String standardOneMenu = ""
- ArrayList< SelectItem > standardRoles = new ArrayList<SelectItem>()
- boolean standardSelected = false
- String tableName = ""
- ArrayList< User > users
- boolean visibleLinks

Statische private Attribute

- static int DEFAULT_CREATE_RIGHT = 15

3.2.3.2 Beschreibung der Konstruktoren und Destruktoren

RoleOverview ()

Definiert in Zeile 221 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedFilteredRights, RoleOverview.assignedQueries, RoleOverview.assignedRights, RoleOverview.attr, RoleOverview.attrObj, RoleOverview.attrRel, RoleOverview.attrTypes, RoleOverview.defaultAccess,

RoleOverview.groups, RoleOverview.hidden, RoleOverview.hidden2,
RoleOverview.hidden3, RoleOverview.hiddenRoleId, RoleOverview.menu,
RoleOverview.meta, RoleOverview.noFilter, RoleOverview.notAssignedGroups,
RoleOverview.notAssignedQueries, RoleOverview.notAssignedUsers,
RoleOverview.objByType, RoleOverview.objects, RoleOverview.objTypes,
RoleOverview.rel, RoleOverview.relations, RoleOverview.relTypes, RoleOverview.role,
RoleOverview.roleId, RoleOverview.roles, RoleOverview.selAccess, RoleOverview.selAttr,
RoleOverview.selAttrObj, RoleOverview.selAttrRel, RoleOverview.selMet,
RoleOverview.selObj, RoleOverview.selObjByType, RoleOverview.selObjType,
RoleOverview.selRel, RoleOverview.selRelTypes, RoleOverview.standardOneMenu,
RoleOverview.standardRoles und RoleOverview.users.

3.2.3.3 Dokumentation der Elementfunktionen

void addGroup () throws Throwable

Fuegt zu einer Rolle eine neue Gruppe hinzu.

Ausnahmebehandlung:

Throwable

Definiert in Zeile 614 der Datei RoleOverview.java.

Benutzt RoleOverview.groups, RoleOverview.notAssignedGroups und RoleOverview.roleId.

void addUser () throws Throwable

Fuegt zu einer Rolle eine neue Gruppe hinzu.

Ausnahmebehandlung:

Throwable

Definiert in Zeile 644 der Datei RoleOverview.java.

Benutzt RoleOverview.notAssignedUsers, RoleOverview.roleId und RoleOverview.users.

boolean alreadyAssigned (long rightId) [private]

Fragt ab, ob eine Funktion schon zugewiesen ist.

Parameter:

rightId

Rückgabe:

true, falls zugewiesen, sonst false

Definiert in Zeile 2549 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedFunc.

Wird benutzt von RoleOverview.listenerSelBundle() und RoleOverview.listenerSelStandard().

String back ()

Setzt alle Werte zurueck.

Rückgabe:

”back”, durch faces-config: Weiterleitung zu roleOverview.jsp.

Definiert in Zeile 541 der Datei RoleOverview.java.

Benutzt RoleOverview.groups, RoleOverview.listLoaded, RoleOverview.sortAscendingOver, RoleOverview.sortColumnOver und RoleOverview.users.

int convertAccess ()

Convertiert das Zugriffsrecht aus checkboxes zu einem Integer

Rückgabe:

int, Zugriffsrecht als Integer

Definiert in Zeile 1676 der Datei RoleOverview.java.

Benutzt RoleOverview.defaultAccess.

Wird benutzt von RoleOverview.saveRights(), RoleOverview.saveRightsByObjType() und RoleOverview.saveRightsByRelType().

String createRole () throws Throwable

Erstellt eine neue Rolle und schreib sie in die Datenbank.

Rückgabe:

”role_edit”, durch die faces-config: Weiterleitung nach roleEdit.jsp

Ausnahmebehandlung:

Throwable

Definiert in Zeile 338 der Datei RoleOverview.java.

Benutzt RoleOverview.listLoaded, RoleOverview.loadFuncRights(), RoleOverview.metaTypeRight, RoleOverview.role, RoleOverview.roleId, RoleOverview.roles und RoleOverview.visibleLinks.

void deleteGroup () throws Throwable

Loescht die gewaehlte Gruppe aus der angezeigten Liste und aus der Verknuepfungstabelle der Datenbank.

Ausnahmebehandlung:**Throwable**

Definiert in Zeile 560 der Datei RoleOverview.java.

Benutzt RoleOverview.groups, RoleOverview.hidden2, RoleOverview.notAssignedGroups und RoleOverview.roleId.

String deleteRight () throws Throwable

Löscht das ausgewählte Recht aus der Rolle

Rückgabe:

JSF forward String

Ausnahmebehandlung:**Throwable**

Definiert in Zeile 2240 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.currentMet, RoleOverview.currentObjTypes, RoleOverview.currentRelTypes, RoleOverview.metaTypeRight, RoleOverview.role, RoleOverview.roleId, RoleOverview.selAttr, RoleOverview.selObj, RoleOverview.selObjByType, RoleOverview.selRel und RoleOverview.selRelTypes.

String deleteRole () throws Throwable

Löscht einer Rolle aus der DB

Rückgabe:

String, JSF forward

Ausnahmebehandlung:**Throwable**

Definiert in Zeile 308 der Datei RoleOverview.java.

Benutzt RoleOverview.role, RoleOverview.roleId, RoleOverview.sortAscendingOver und RoleOverview.sortColumnOver.

void deleteSelectedFunctions () throws Throwable

Entfernt mehrere Funktionen aus der Verknuepfung der Rolle.

Ausnahmebehandlung:**Throwable**

Definiert in Zeile 2698 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedFunc, RoleOverview.bundleId, RoleOverview.filtAssignedRights, RoleOverview.noFilter, RoleOverview.role, RoleOverview.roleId, RoleOverview.selectableRights und RoleOverview.standardOneMenu.

String deleteSelectedRights () throws Throwable

Löscht die durch check boxes ausgewählte Rechte aus der Rolle

Rückgabe:

JSF forward String

Ausnahmebehandlung:

Throwable

Definiert in Zeile 2281 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.currentMet, RoleOverview.currentObjTypes, RoleOverview.currentRelTypes, RoleOverview.metaTypeRight, RoleOverview.role, RoleOverview.roleId, RoleOverview.selAttr, RoleOverview.selObj, RoleOverview.selObjByType und RoleOverview.selRel.

void deleteUser () throws Throwable

Loescht den gewaehlten Nutzer aus der angezeigten Liste und aus der Verknuepfungstabelle der Datenbank.

Ausnahmebehandlung:

Throwable

Definiert in Zeile 587 der Datei RoleOverview.java.

Benutzt RoleOverview.hidden3, RoleOverview.notAssignedUsers, RoleOverview.roleId und RoleOverview.users.

String editAssignedRole () throws Throwable

Wird von einer Gruppen oder Nutzer Detailseite aufgerufen. Holt die gewuenschte Rolle aus der Datenbank und stellt so die notwendigen Informationen bereit, roleEdit.jsp aufzubauen.

Rückgabe:

"role_edit", durch die faces-config: Weiterleitung nach roleEdit.jsp

Ausnahmebehandlung:

Throwable

Definiert in Zeile 435 der Datei RoleOverview.java.

Benutzt RoleOverview.listLoaded, RoleOverview.loadFuncRights(), RoleOverview.menu, RoleOverview.metaTypeRight, RoleOverview.role, RoleOverview.roleId, ManageMenu.setSortAscendingAs(), ManageMenu.setSortAscendingAs2(), ManageMenu.setSortAscendingNoAs(), ManageMenu.setSortAscendingNoAs2(), ManageMenu.setSortColumnAs(), ManageMenu.setSortColumnAs2(), ManageMenu.setSortColumnNoAs(), ManageMenu.setSortColumnNoAs2(), RoleOverview.sortQueriesAssignedAscending, RoleOverview.sortQueriesAssignedColumn, RoleOverview.sortQueriesNotAssignedAscending, RoleOverview.sortQueriesNotAssignedColumn und RoleOverview.visibleLinks.

void editRole () throws Throwable

saveRole speichert die neuen Werte, Name und Description, in der Datenbank.

Ausnahmebehandlung:**Throwable**

Definiert in Zeile 475 der Datei RoleOverview.java.

Benutzt RoleOverview.role und RoleOverview.roleId.

IDalRole generateDalRoleObject () throws ProxyException, Throwable [private]

Erzeugt fuer die Datenbankzugriffe mit der DalRole-Schnittstelle das notwendige Objekt

Rückgabe:

ein DalRole Objekt

Ausnahmebehandlung:**ProxyException****Throwable**

Definiert in Zeile 3123 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.listenerCreateAttrTyp(), RoleOverview.listenerCreateMeta(), RoleOverview.listenerCreateObjType(), RoleOverview.listenerCreateRelType(), RoleOverview.listenerCreateStatus() und RoleOverview.listenerCreateTime().

ArrayList<AuthInfo> getAssignedFilteredRights ()

Definiert in Zeile 975 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedFilteredRights.

LinkedList<Query> getAssignedQueries () throws Throwable

Liefert die zugewiesenen Queries

Rückgabe:

eine LinkedList mit Query-Objekten

Ausnahmebehandlung:

Throwable

Definiert in Zeile 2841 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedQueries und RoleOverview.roleId.

ArrayList<AuthInfo> getAssignedRights () throws Throwable

Liefert alle Rechte, die der Rolle zugeordnet sind

Rückgabe:

ArrayList<AuthInfo>, Liste mit AuthInfo Objekten, die Rechte enthalten

Ausnahmebehandlung:

Throwable

Definiert in Zeile 890 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.createRight, RoleOverview.listLoaded, RoleOverview.metaTypeRight, RoleOverview.roleId, RoleOverview.selectAllRights und RoleOverview.setCreateAccessDefault().

Wird benutzt von RoleOverview.getSelMet(), RoleOverview.getSelObjByType(), RoleOverview.getSelObjType() und RoleOverview.getSelRelTypes().

ArrayList<AuthInfo> getAttr ()

Definiert in Zeile 999 der Datei RoleOverview.java.

Benutzt RoleOverview.attr.

AuthInfo getAttrInfoById (int id)

Sucht AuthInfo Objekt zu einem Attribut anhand einer Id

return AuthInfo

Definiert in Zeile 2401 der Datei RoleOverview.java.

Benutzt RoleOverview.attr.

Wird benutzt von RoleOverview.saveRights().

ArrayList<AuthInfo> getAttrObj ()

Definiert in Zeile 1066 der Datei RoleOverview.java.

Benutzt RoleOverview.attrObj.

ArrayList<AuthInfo> getAttrRel ()

Definiert in Zeile 1058 der Datei RoleOverview.java.

Benutzt RoleOverview.attrRel.

ArrayList<AuthInfo> getAttrTypes ()

Definiert in Zeile 1010 der Datei RoleOverview.java.

Benutzt RoleOverview.attrTypes.

ArrayList<SelectItem> getBundleList ()

Definiert in Zeile 2735 der Datei RoleOverview.java.

Benutzt RoleOverview.bundleList.

String getBundleOneList ()

Definiert in Zeile 2753 der Datei RoleOverview.java.

Benutzt RoleOverview.bundleOneList.

String getBundleOneMenu ()

Definiert in Zeile 2766 der Datei RoleOverview.java.

Benutzt RoleOverview.bundleOneMenu.

ArrayList<Module> getBundles ()

Definiert in Zeile 2760 der Datei RoleOverview.java.

Benutzt RoleOverview.bundles.

int getColumnNumbers ()

Definiert in Zeile 2448 der Datei RoleOverview.java.

Benutzt RoleOverview.metaTypeRight.

String [] getCurrentAttr ()

Definiert in Zeile 1095 der Datei RoleOverview.java.

Benutzt RoleOverview.currentAttr.

String [] getCurrentAttrObj ()

Definiert in Zeile 1158 der Datei RoleOverview.java.

Benutzt RoleOverview.currentAttrObj.

String [] getCurrentAttrRel ()

Definiert in Zeile 1150 der Datei RoleOverview.java.

Benutzt RoleOverview.currentAttrRel.

String [] getCurrentAttrTypes ()

Definiert in Zeile 1110 der Datei RoleOverview.java.

Benutzt RoleOverview.currentAttrTypes.

String [] getCurrentMet ()

Definiert in Zeile 1083 der Datei RoleOverview.java.

Benutzt RoleOverview.currentMet.

String [] getCurrentObj ()

Definiert in Zeile 1089 der Datei RoleOverview.java.

Benutzt RoleOverview.currentObj.

String [] getCurrentObjByType ()

Definiert in Zeile 1142 der Datei RoleOverview.java.

Benutzt RoleOverview.currentObjByType.

String [] getCurrentObjTypes ()

Definiert in Zeile 1102 der Datei RoleOverview.java.

Benutzt RoleOverview.currentObjTypes.

String [] getCurrentRel ()

Definiert in Zeile 1134 der Datei RoleOverview.java.

Benutzt RoleOverview.currentRel.

String [] getCurrentRelTypes ()

Definiert in Zeile 1126 der Datei RoleOverview.java.

Benutzt RoleOverview.currentRelTypes.

String [] getDefaultAccess ()

Definiert in Zeile 1118 der Datei RoleOverview.java.

Benutzt RoleOverview.defaultAccess.

ArrayList<View> getFiltAssignedRights ()

Definiert in Zeile 2785 der Datei RoleOverview.java.

Benutzt RoleOverview.filtAssignedRights.

ArrayList<Group> getGroups () throws Throwable

Rückgabe:

groups, Gruppen der Rolle

Ausnahmebehandlung:

Throwable

Definiert in Zeile 705 der Datei RoleOverview.java.

Benutzt RoleOverview.getGroupsByRole() und RoleOverview.groups.

ArrayList<Group> getGroupsByRole () throws Throwable

Laedt aus der DB alle Gruppen die diese Rolle verwenden.

Rückgabe:

ArrayList<Group> temp, die Liste mit den Gruppen.

Ausnahmebehandlung:

Throwable

Definiert in Zeile 511 der Datei RoleOverview.java.

Benutzt RoleOverview.roleId.

Wird benutzt von RoleOverview.getGroups().

HtmlInputHidden getHidden ()

Definiert in Zeile 769 der Datei RoleOverview.java.

Benutzt RoleOverview.hidden.

HtmlInputHidden getHidden2 ()

Definiert in Zeile 868 der Datei RoleOverview.java.

Benutzt RoleOverview.hidden2.

HtmlInputHidden getHidden3 ()

Definiert in Zeile 876 der Datei RoleOverview.java.

Benutzt RoleOverview.hidden3.

HtmlInputHidden getHiddenRoleId ()

Definiert in Zeile 2475 der Datei RoleOverview.java.

Benutzt RoleOverview.hiddenRoleId.

HtmlInputHidden getHiddenView ()

Definiert in Zeile 2803 der Datei RoleOverview.java.

Benutzt RoleOverview.hiddenView.

ManageMenu getMenu ()

Definiert in Zeile 2469 der Datei RoleOverview.java.

Benutzt RoleOverview.menu.

ArrayList<AuthInfo> getMeta ()

Definiert in Zeile 996 der Datei RoleOverview.java.

Benutzt RoleOverview.meta.

AuthInfo getMetInfoById (int id)

Sucht AuthInfo Objekt zu einem Metamodell anhand einer Id

Rückgabe:

AuthInfo

Definiert in Zeile 2317 der Datei RoleOverview.java.

Benutzt RoleOverview.meta.

Wird benutzt von RoleOverview.saveRights().

ArrayList<Group> getNotAssignedGroups () throws Throwable

Liefert alle Gruppen, die der Rolle nicht zugeordnet sind

Rückgabe:

ArrayList<Group>, List mit Group Objekten

Ausnahmebehandlung:**Throwable**

Definiert in Zeile 837 der Datei RoleOverview.java.

Benutzt RoleOverview.groups und RoleOverview.notAssignedGroups.

LinkedList<Query> getNotAssignedQueries () throws Throwable

Liefert die nicht zugewiesenen Queries

Rückgabe:

eine LinkedList mit Query-Objekten

Ausnahmebehandlung:**Throwable**

Definiert in Zeile 2855 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedQueries und RoleOverview.notAssignedQueries.

ArrayList<User> getNotAssignedUsers () throws Throwable

Liefert alle User, die der Rolle nicht zugeordnet sind

Rückgabe:

ArrayList<User>, Liste mit User Objekten

Ausnahmebehandlung:**Throwable**

Definiert in Zeile 808 der Datei RoleOverview.java.

Benutzt RoleOverview.notAssignedUsers und RoleOverview.users.

AuthInfo getObjAttributeInfoById (int id)

Sucht AuthInfo Objekt zu einem Attribut anhand einer Id (Tab Objekttyp)

return AuthInfo

Definiert in Zeile 2413 der Datei RoleOverview.java.

Benutzt RoleOverview.attrObj.

Wird benutzt von RoleOverview.saveRightsByObjType().

ArrayList<AuthInfo> getObjByType ()

Definiert in Zeile 1042 der Datei RoleOverview.java.

Benutzt RoleOverview.objByType.

AuthInfo getObjByTypeInfoById (int id)

Sucht AuthInfo Objekt zu einem Objekt anhand einer Id

Rückgabe:

AuthInfo

Definiert in Zeile 2353 der Datei RoleOverview.java.

Benutzt RoleOverview.objByType.

Wird benutzt von RoleOverview.saveRightsByObjType().

ArrayList<AuthInfo> getObjects ()

Definiert in Zeile 993 der Datei RoleOverview.java.

Benutzt RoleOverview.objects.

AuthInfo getObjInfoById (int id)

Sucht AuthInfo Objekt zu einem Objekt anhand einer Id

Rückgabe:

AuthInfo

Definiert in Zeile 2329 der Datei RoleOverview.java.

Benutzt RoleOverview.objects.

Wird benutzt von RoleOverview.saveRights().

AuthInfo getObjTypeInfoById (int id)

Sucht AuthInfo Objekt zu einem Objekttyp anhand einer Id

Rückgabe:

AuthInfo

Definiert in Zeile 2365 der Datei RoleOverview.java.

Benutzt RoleOverview.objTypes.

Wird benutzt von RoleOverview.saveRightsByObjType().

ArrayList<AuthInfo> getObjTypes ()

Definiert in Zeile 1002 der Datei RoleOverview.java.

Benutzt RoleOverview.objTypes.

ArrayList<AuthInfo> getRel ()

Definiert in Zeile 1034 der Datei RoleOverview.java.

Benutzt RoleOverview.rel.

AuthInfo getRelationByld (int id)

Sucht AuthInfo Objekt zu einer Relation, die anhand eines Metamodells ausgewählt wurde

Rückgabe:

AuthInfo

Definiert in Zeile 2342 der Datei RoleOverview.java.

Benutzt RoleOverview.relations.

Wird benutzt von RoleOverview.saveRights().

ArrayList<AuthInfo> getRelations ()

Definiert in Zeile 1050 der Datei RoleOverview.java.

Benutzt RoleOverview.relations.

AuthInfo getRelAttributeInfoByld (int id)

Sucht AuthInfo Objekt zu einem Attribut anhand einer Id (Tab Relationstyp)

return AuthInfo

Definiert in Zeile 2425 der Datei RoleOverview.java.

Benutzt RoleOverview.attrRel.

Wird benutzt von RoleOverview.saveRightsByRelType().

AuthInfo getRelInfoByld (int id)

Sucht AuthInfo Objekt zu einer Relation anhand einer Id

Rückgabe:

AuthInfo

Definiert in Zeile 2389 der Datei RoleOverview.java.

Benutzt RoleOverview.rel.

Wird benutzt von RoleOverview.saveRightsByRelType().

AuthInfo getRelTypeInfoByld (int id)

Sucht AuthInfo Objekt zu einem Relationstyp anhand einer Id

Rückgabe:

AuthInfo

Definiert in Zeile 2377 der Datei RoleOverview.java.

Benutzt RoleOverview.relTypes.

Wird benutzt von RoleOverview.saveRightsByRelType().

ArrayList<AuthInfo> getRelTypes ()

Definiert in Zeile 1026 der Datei RoleOverview.java.

Benutzt RoleOverview.relTypes.

String getRightTypeChoice ()

Definiert in Zeile 2433 der Datei RoleOverview.java.

Benutzt RoleOverview.rightTypeChoice.

Role getRole ()

Gibt die aktuelle Rolle zurück

Rückgabe:

role, Role

Definiert in Zeile 798 der Datei RoleOverview.java.

Benutzt RoleOverview.role.

int getRoleId ()

Rückgabe:

id, ID der aktuellen Rolle

Definiert in Zeile 781 der Datei RoleOverview.java.

Benutzt RoleOverview.roleId.

ArrayList<Role> getRoles () throws Throwable

Laed alle Rollen aus der Datenbank

Rückgabe:

roles, ArrayList<Role> Liste mit allen Rollen

Ausnahmebehandlung:

Throwable

Definiert in Zeile 742 der Datei RoleOverview.java.

Benutzt RoleOverview.menu, RoleOverview.roles, ManageMenu.setSortAscendingAs(), ManageMenu.setSortAscendingAs2(), ManageMenu.setSortAscendingNoAs(), ManageMenu.setSortAscendingNoAs2(), ManageMenu.setSortColumnAs(), ManageMenu.setSortColumnAs2(), ManageMenu.setSortColumnNoAs() und ManageMenu.setSortColumnNoAs2().

ArrayList<SelectedItem> getSelAccess () throws Throwable

Liefert eine Liste mit den Zugriffsrechten

Rückgabe:

ArrayList<SelectedItem>, Liste mit den Zugriffsrechten

Ausnahmebehandlung:

Throwable

Definiert in Zeile 1364 der Datei RoleOverview.java.

Benutzt RoleOverview.selAccess.

ArrayList<SelectedItem> getSelAttr () throws Throwable

Liefert eine Liste mit den Attributen, die zu den ausgewählten Objekten und Relationen gehören

Rückgabe:

ArrayList<SelectedItem>, Liste mit den Attributen

Ausnahmebehandlung:

Throwable

Definiert in Zeile 1274 der Datei RoleOverview.java.

Benutzt RoleOverview.attr, RoleOverview.ATTRIBUTE_RIGHT, RoleOverview.currentObj, RoleOverview.isInAssignedRights(), RoleOverview.OBJECT_RIGHT, RoleOverview.objSelected und RoleOverview.selAttr.

ArrayList<SelectedItem> getSelAttrObj () throws Throwable

Liefert eine Liste mit den Attributen, die zu den ausgewählten Objekten gehören (Objektyp Tab)

Rückgabe:

ArrayList<SelectedItem>, Liste mit den Attributen

Ausnahmebehandlung:

Throwable

Definiert in Zeile 1534 der Datei RoleOverview.java.

Benutzt RoleOverview.ATTRIBUTE_RIGHT, RoleOverview.attrObj, RoleOverview.currentObjByType, RoleOverview.isInAssignedRights(), RoleOverview.objByTypeSelected und RoleOverview.selAttrObj.

ArrayList<SelectItem> getSelAttrRel () throws Throwable

Liefert eine Liste mit den Attributen, die zu den ausgewählten Relationen gehören (Relationstyp Tab)

Rückgabe:

ArrayList<SelectItem>, Liste mit den Attributen

Ausnahmebehandlung:

Throwable

Definiert in Zeile 1498 der Datei RoleOverview.java.

Benutzt RoleOverview.ATTRIBUTE_RIGHT, RoleOverview.attrRel, RoleOverview.currentRel, RoleOverview.isInAssignedRights(), RoleOverview.rel, RoleOverview.relByTypeSelected und RoleOverview.selAttrRel.

ArrayList<SelectItem> getSelectableRights ()

Definiert in Zeile 2741 der Datei RoleOverview.java.

Benutzt RoleOverview.selectableRights.

ArrayList<String> getSelectedRights ()

Definiert in Zeile 2747 der Datei RoleOverview.java.

Benutzt RoleOverview.selectedRights.

ArrayList<SelectItem> getSelMet () throws Throwable

Zeigt alle Metamodelle, für die der Benutzer keine Rechte besitzt

Rückgabe:

selMet, ArrayList<SelectItem> Liste mit den Metamodelle

Ausnahmebehandlung:

Throwable

Definiert in Zeile 1180 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.getAssignedRights(), RoleOverview.isInAssignedRights(), RoleOverview.meta, RoleOverview.METAMODEL_RIGHT und RoleOverview.selMet.

ArrayList<SelectItem> getSelObj () throws Throwable

Liefert alle Objekte und Relationen, die zu den Ausgewählten Metamodellen gehören

Rückgabe:

ArrayList<SelectItem>, Liste mit Objekten und Relationen

Ausnahmebehandlung:

Throwable

Definiert in Zeile 1219 der Datei RoleOverview.java.

Benutzt RoleOverview.currentMet, RoleOverview.isInAssignedRights(), RoleOverview.metaSelected, RoleOverview.OBJECT_RIGHT, RoleOverview.objects, RoleOverview.RELATION_RIGHT, RoleOverview.relations und RoleOverview.selObj.

ArrayList<SelectItem> getSelObjByType () throws Throwable

Liefert eine Liste mit den Objekten, die zu den ausgewählten Objekttypen gehören

Rückgabe:

ArrayList<SelectItem>, Liste mit den Objekten

Ausnahmebehandlung:

Throwable

Definiert in Zeile 1463 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.currentObjTypes, RoleOverview.getAssignedRights(), RoleOverview.isInAssignedRights(), RoleOverview.objByType, RoleOverview.OBJECT_RIGHT, RoleOverview.objects, RoleOverview.objTypeSelected und RoleOverview.selObjByType.

ArrayList<SelectItem> getSelObjType () throws Throwable

Liefert eine Liste mit den Objekttypen

Rückgabe:

ArrayList<SelectItem>, Liste mit den Objekttypen

Ausnahmebehandlung:

Throwable

Definiert in Zeile 1321 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.getAssignedRights(), RoleOverview.isInAssignedRights(), RoleOverview.OBJECT_TYPE_RIGHT, RoleOverview.objTypes und RoleOverview.selObjType.

ArrayList<SelectItem> getSelRel () throws Throwable

Liefert eine Liste mit den Relatione, die zu den asugewählten Relationstypen gehören

Rückgabe:

ArrayList<SelectItem>, Liste mit den Relationen

Ausnahmebehandlung:

Throwable

Definiert in Zeile 1398 der Datei RoleOverview.java.

Benutzt RoleOverview.currentRelTypes, RoleOverview.isInAssignedRights(), RoleOverview.rel, RoleOverview.RELATION_RIGHT, RoleOverview.relations, RoleOverview.relTypeSelected und RoleOverview.selRel.

ArrayList<SelectItem> getSelRelTypes () throws Throwable

Liefert eine Liste mit den Relationstypen

Rückgabe:

ArrayList<SelectItem>, Liste mit den Relationstypen

Ausnahmebehandlung:

Throwable

Definiert in Zeile 1433 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.getAssignedRights(), RoleOverview.isInAssignedRights(), RoleOverview.RELATION_TYPE_RIGHT, RoleOverview.relTypes und RoleOverview.selRelTypes.

String getSortColumnOver ()

Definiert in Zeile 2457 der Datei RoleOverview.java.

Benutzt RoleOverview.sortColumnOver.

String getSortQueriesAssignedColumn ()

Definiert in Zeile 2809 der Datei RoleOverview.java.

Benutzt RoleOverview.sortQueriesAssignedColumn.

String getSortQueriesNotAssignedColumn ()

Definiert in Zeile 2821 der Datei RoleOverview.java.

Benutzt RoleOverview.sortQueriesNotAssignedColumn.

String getStandardOneMenu ()

Definiert in Zeile 2778 der Datei RoleOverview.java.

Benutzt RoleOverview.standardOneMenu.

ArrayList<SelectedItem> getStandardRoles ()

Definiert in Zeile 2772 der Datei RoleOverview.java.

Benutzt RoleOverview.standardRoles.

String getTableName ()

Gibt den Namen der Tabelle für Rechte zurück (Rechte über Daten/Rechte über Metamodelle)

Rückgabe:

tableData

Definiert in Zeile 953 der Datei RoleOverview.java.

Benutzt RoleOverview.metaTypeRight und RoleOverview.tableName.

ArrayList<User> getUsers () throws Throwable**Rückgabe:**

users, Benutzer der Gruppe

Ausnahmebehandlung:

Throwable

Definiert in Zeile 723 der Datei RoleOverview.java.

Benutzt RoleOverview.getUsersByRole() und RoleOverview.users.

ArrayList<User> getUsersByRole () throws Throwable

Laedt aus der DB alle Nutzern die diese Rolle verwenden.

Rückgabe:

ArrayList<User> temp, die Liste mit den Nutzern.

Ausnahmebehandlung:

Throwable

Definiert in Zeile 527 der Datei RoleOverview.java.

Benutzt RoleOverview.roleId.

Wird benutzt von RoleOverview.getUsers().

void goAssigned (ActionEvent ae)

Definiert in Zeile 406 der Datei RoleOverview.java.

Benutzt RoleOverview.menu, ManageMenu.setSortAscendingAs(),
ManageMenu.setSortAscendingAs2(), ManageMenu.setSortAscendingNoAs(),
ManageMenu.setSortAscendingNoAs2(), ManageMenu.setSortColumnAs(),
ManageMenu.setSortColumnAs2(), ManageMenu.setSortColumnNoAs(),
ManageMenu.setSortColumnNoAs2(), RoleOverview.sortQueriesAssignedAscending,
RoleOverview.sortQueriesAssignedColumn,
RoleOverview.sortQueriesNotAssignedAscending und
RoleOverview.sortQueriesNotAssignedColumn.

String goToCreateNewRole ()

Leitet auf die Seite zur Erstellung einer neuen Rolle

Rückgabe:

String, forward für JSF

Definiert in Zeile 498 der Datei RoleOverview.java.

Benutzt RoleOverview.role.

String goToDeleteFromEdit ()

delete-Aufruf von role_edit aus.

Rückgabe:

String, für Navigation mit faces-config

Definiert in Zeile 283 der Datei RoleOverview.java.

String goToDeleteFromOverview ()

delete-Aufruf von role_overview aus.

Rückgabe:

String, für Navigation mit faces-config

Definiert in Zeile 292 der Datei RoleOverview.java.

Benutzt RoleOverview.hidden, RoleOverview.role, RoleOverview.roleId und RoleOverview.roles.

String goToEditRole () throws Throwable

Sucht die gewünschte Rolle aus der Liste und stellt so die notwendigen Informationen bereit, roleEdit.jsp aufzubauen.

Rückgabe:

”role_edit”, durch die faces-config: Weiterleitung nach roleEdit.jsp

Ausnahmebehandlung:

Throwable

Definiert in Zeile 368 der Datei RoleOverview.java.

Benutzt RoleOverview.listLoaded, RoleOverview.loadFuncRights(), RoleOverview.menu, RoleOverview.metaTypeRight, RoleOverview.rightTypeChoice, RoleOverview.role, RoleOverview.roleId, RoleOverview.roles, ManageMenu.setSortAscendingAs(), ManageMenu.setSortAscendingAs2(), ManageMenu.setSortAscendingNoAs(), ManageMenu.setSortAscendingNoAs2(), ManageMenu.setSortColumnAs(), ManageMenu.setSortColumnAs2(), ManageMenu.setSortColumnNoAs(), ManageMenu.setSortColumnNoAs2(), RoleOverview.sortQueriesAssignedAscending, RoleOverview.sortQueriesAssignedColumn, RoleOverview.sortQueriesNotAssignedAscending, RoleOverview.sortQueriesNotAssignedColumn und RoleOverview.visibleLinks.

boolean isBundleSelected ()

Definiert in Zeile 2791 der Datei RoleOverview.java.

Benutzt RoleOverview.bundleSelected.

boolean isCreateAttrTyp ()

Definiert in Zeile 3021 der Datei RoleOverview.java.

Benutzt RoleOverview.createRight.

boolean isCreateMeta ()

Definiert in Zeile 2952 der Datei RoleOverview.java.

Benutzt RoleOverview.createRight.

boolean isCreateObjType ()

Definiert in Zeile 2964 der Datei RoleOverview.java.

Benutzt RoleOverview.createRight.

boolean isCreateRelType ()

Definiert in Zeile 2958 der Datei RoleOverview.java.

Benutzt RoleOverview.createRight.

boolean isCreateStatus ()

Definiert in Zeile 3055 der Datei RoleOverview.java.

Benutzt RoleOverview.createRight.

boolean isCreateTime ()

Definiert in Zeile 3038 der Datei RoleOverview.java.

Benutzt RoleOverview.createRight.

boolean isInAssignedRights (String value, String object)

Überprüft, ob das Recht schon vorhanden ist

Parameter:

String value, Id des Rechtes

String object, Typ des Rechts

Rückgabe:

boolean

Definiert in Zeile 1657 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights.

Wird benutzt von RoleOverview.getSelAttr(), RoleOverview.getSelAttrObj(), RoleOverview.getSelAttrRel(), RoleOverview.getSelMet(), RoleOverview.getSelObj(), RoleOverview.getSelObjByType(), RoleOverview.getSelObjType(), RoleOverview.getSelRel() und RoleOverview.getSelRelTypes().

boolean isMetaTypeRight ()

Definiert in Zeile 2440 der Datei RoleOverview.java.

Benutzt RoleOverview.metaTypeRight.

boolean isSortAscendingOver ()

Definiert in Zeile 2463 der Datei RoleOverview.java.

Benutzt RoleOverview.sortAscendingOver.

boolean isSortQueriesAssignedAscending ()

Definiert in Zeile 2815 der Datei RoleOverview.java.

Benutzt RoleOverview.sortQueriesAssignedAscending.

boolean isSortQueriesNotAssignedAscending ()

Definiert in Zeile 2827 der Datei RoleOverview.java.

Benutzt RoleOverview.sortQueriesNotAssignedAscending.

boolean isStandardSelected ()

Definiert in Zeile 2797 der Datei RoleOverview.java.

Benutzt RoleOverview.standardSelected.

boolean isVisibleLinks ()

Definiert in Zeile 979 der Datei RoleOverview.java.

Benutzt RoleOverview.visibleLinks.

void listenerCreateAttrTyp (ValueChangeEvent event) throws ProxyException, Throwable

Definiert in Zeile 3026 der Datei RoleOverview.java.

Benutzt RoleOverview.createRight, RoleOverview.DEFAULT_CREATE_RIGHT, RoleOverview.generateDalRoleObject() und RoleOverview.roleId.

void listenerCreateMeta (ValueChangeEvent event) throws ProxyException, Throwable

Definiert in Zeile 3078 der Datei RoleOverview.java.

Benutzt RoleOverview.createRight, RoleOverview.DEFAULT_CREATE_RIGHT, RoleOverview.generateDalRoleObject() und RoleOverview.roleId.

void listenerCreateObjType (ValueChangeEvent event) throws ProxyException, Throwable

Definiert in Zeile 3104 der Datei RoleOverview.java.

Benutzt RoleOverview.createRight, RoleOverview.DEFAULT_CREATE_RIGHT, RoleOverview.generateDalRoleObject() und RoleOverview.roleId.

void listenerCreateRelType (ValueChangeEvent event) throws ProxyException, Throwable

Definiert in Zeile 3091 der Datei RoleOverview.java.

Benutzt RoleOverview.createRight, RoleOverview.DEFAULT_CREATE_RIGHT, RoleOverview.generateDalRoleObject() und RoleOverview.roleId.

void listenerCreateStatus (ValueChangeEvent event) throws ProxyException, Throwable

Definiert in Zeile 3060 der Datei RoleOverview.java.

Benutzt RoleOverview.createRight, RoleOverview.DEFAULT_CREATE_RIGHT, RoleOverview.generateDalRoleObject() und RoleOverview.roleId.

void listenerCreateTime (ValueChangeEvent event) throws ProxyException, Throwable

Definiert in Zeile 3043 der Datei RoleOverview.java.

Benutzt RoleOverview.createRight, RoleOverview.DEFAULT_CREATE_RIGHT, RoleOverview.generateDalRoleObject() und RoleOverview.roleId.

void listenerSelBundle (ValueChangeEvent event)

Faengt einen Event ab, der geworfen wird, wenn ein Bundle ausgewaehlt wurde. Die Funktionen eines Bundles, welche noch nicht zugewiesen wurden, werden dem Benutzer so zur Verfuegung gestellt.

Parameter:

event

Definiert in Zeile 2521 der Datei RoleOverview.java.

Benutzt RoleOverview.alreadyAssigned(), RoleOverview.bundleId, RoleOverview.bundles, RoleOverview.bundleSelected, RoleOverview.noFilter, RoleOverview.selectableRights und RoleOverview.standardOneMenu.

void listenerSelStandard (ValueChangeEvent event)

Faengt den Event ab, der geworfen wurde, wenn eine Zachman-Rolle ausgewaehlt wurde. Hier werden die Rechte herausgefiltert, welche nicht dazugehoeren.

Parameter:

event

Definiert in Zeile 2564 der Datei RoleOverview.java.

Benutzt RoleOverview.alreadyAssigned(), RoleOverview.assignedFunc, RoleOverview.bundleId, RoleOverview.bundles, RoleOverview.filtAssignedRights, RoleOverview.selectableRights, RoleOverview.standardOneMenu und RoleOverview.standardSelected.

void loadFuncRights () throws Throwable

Kapselt Einstellungen fuer die Funktionen-Vergabe. Dies sind unter anderem: Laden der Bundles und ihrer Funktionen aus der Datenbank, Laden der zugewiesenen Funktionen der Rolle und das Fuellen der Selectitemliste.

Ausnahmebehandlung:**Throwable**

Definiert in Zeile 2489 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedFunc, RoleOverview.bundleId, RoleOverview.bundleList, RoleOverview.bundleOneList, RoleOverview.bundleOneMenu, RoleOverview.bundles, RoleOverview.bundleSelected, RoleOverview.filterAssignedRights, RoleOverview.roleId, RoleOverview.selectableRights und RoleOverview.standardSelected.

Wird benutzt von RoleOverview.createRole(), RoleOverview.editAssignedRole() und RoleOverview.goToEditRole().

void modellSelected (ValueChangeEvent event)

Definiert in Zeile 1572 der Datei RoleOverview.java.

Benutzt RoleOverview.currentMet, RoleOverview.metaSelected, RoleOverview.objSelected, RoleOverview.selAttr und RoleOverview.selObj.

void objByTypeSelected (ValueChangeEvent event)

Definiert in Zeile 1622 der Datei RoleOverview.java.

Benutzt RoleOverview.currentObjByType, RoleOverview.objByTypeSelected und RoleOverview.selAttrObj.

void objSelected (ValueChangeEvent event)

Definiert in Zeile 1593 der Datei RoleOverview.java.

Benutzt RoleOverview.currentAttr, RoleOverview.currentObj, RoleOverview.objSelected und RoleOverview.selAttr.

void objTypeSelected (ValueChangeEvent event)

Definiert in Zeile 1583 der Datei RoleOverview.java.

Benutzt RoleOverview.currentObjTypes, RoleOverview.objTypeSelected und RoleOverview.selObjByType.

void queryAssign () throws Throwable

Weist einer Rolle eine Query zu

Ausnahmebehandlung:

Throwable

Definiert in Zeile 2883 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedQueries, RoleOverview.notAssignedQueries, RoleOverview.role und RoleOverview.roleId.

void queryRemoveOne () throws Throwable

Entfernt eine Query aus der Zuweisungstabelle zwischen Queries und Rollen

Ausnahmebehandlung:

Throwable

Definiert in Zeile 2923 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedQueries, RoleOverview.hidden, RoleOverview.notAssignedQueries, RoleOverview.role und RoleOverview.roleId.

void relByTypeSelected (ValueChangeEvent event)

Definiert in Zeile 1613 der Datei RoleOverview.java.

Benutzt RoleOverview.currentRel, RoleOverview.relByTypeSelected und RoleOverview.selAttrRel.

void relTypeSelected (ValueChangeEvent event)

Definiert in Zeile 1604 der Datei RoleOverview.java.

Benutzt RoleOverview.currentRelTypes, RoleOverview.relTypeSelected und RoleOverview.selRel.

void removeOneAssignment () throws Throwable

Entfernt eine View.

Ausnahmebehandlung:

Throwable

Definiert in Zeile 2658 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedFunc, RoleOverview.bundleId, RoleOverview.filtAssignedRights, RoleOverview.hiddenView, RoleOverview.noFilter, RoleOverview.role, RoleOverview.roleId, RoleOverview.selectableRights und RoleOverview.standardOneMenu.

void resolveMetamodelDependencies (int modelId, int access) throws Throwable

Überprüft die Abhängigkeiten zwischen den zu speichernden Metamodellen und schon gespeicherten Objekten und Attributen (siehe Entwurf Rechtehierarchie).

Parameter:

modelId,Id des Metamodells um die Zugehörigen Attribute auszulesen
access,Zugriffsrecht

Ausnahmebehandlung:

Throwable

Definiert in Zeile 2004 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.ATTRIBUTE_RIGHT, RoleOverview.metaTypeRight, RoleOverview.OBJECT_RIGHT, RoleOverview.objects, RoleOverview.RELATION_RIGHT, RoleOverview.relations und RoleOverview.roleId.

Wird benutzt von RoleOverview.saveRights().

void resolveObjByTypeDependencies (int access) throws Throwable

Überprüft, ob es Abhängigkeiten zwischen den Objekten (aus dem Objekttyp Tab) und schon gespeicherten Attributenrechten gibt.

Parameter:

access,Zugriffsrecht

Ausnahmebehandlung:

ProxyException

DatabaseException

Definiert in Zeile 2213 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.attr, RoleOverview.ATTRIBUTE_RIGHT, RoleOverview.attrObj, RoleOverview.metaTypeRight und RoleOverview.roleId.

Wird benutzt von RoleOverview.saveRightsByObjType().

void resolveObjectDependencies (int access) throws Throwable

Überprüft die Abhängigkeiten zwischen den zu speichernden Objekten/Relationen und schon gespeicherten Attributen (siehe Entwurf Rechtehierarchie).

Parameter:

access,Zugriffsrecht

Ausnahmebehandlung:

Throwable

Definiert in Zeile 2056 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.attr, RoleOverview.metaTypeRight und RoleOverview.roleId.

Wird benutzt von RoleOverview.saveRights().

void resolveObjecttypeDependencies (int access) throws Throwable

Überprüft die Abhängigkeiten zwischen den zu speichernden Objekttypen und schon gespeicherten Objekten und Attributen.

Parameter:

access, Zugriffsrecht

Ausnahmebehandlung:

Throwable

Definiert in Zeile 2162 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.attr, RoleOverview.ATTRIBUTE_RIGHT, RoleOverview.metaTypeRight, RoleOverview.objByType, RoleOverview.OBJECT_RIGHT und RoleOverview.roleId.

Wird benutzt von RoleOverview.saveRightsByObjType().

void resolveRelationtypeDependencies (int access) throws Throwable

Überprüft die Abhängigkeiten zwischen den zu speichernden Relationstypen und schon gespeicherten Relationen und Attributen.

Parameter:

access, Zugriffsrecht

Ausnahmebehandlung:

Throwable

Definiert in Zeile 2083 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.ATTRIBUTE_RIGHT, RoleOverview.metaTypeRight, RoleOverview.rel, RoleOverview.RELATION_RIGHT und RoleOverview.roleId.

Wird benutzt von RoleOverview.saveRightsByRelType().

void resolveRelByTypeDependencies (int access) throws Throwable

Überprüft, ob es Abhängigkeiten zwischen den Relationen (aus Relationstyp Tab) und den gespeicherten Attributenrechten gibt

Parameter:

access,Zugriffsrecht

Ausnahmebehandlung:

ProxyException

DatabaseException

Definiert in Zeile 2134 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.attr, RoleOverview.ATTRIBUTE_RIGHT, RoleOverview.attrRel, RoleOverview.metaTypeRight und RoleOverview.roleId.

Wird benutzt von RoleOverview.saveRightsByRelType().

String saveAccess () throws Throwable

Speichert die Zugriffsmöglichkeiten auf die Daten. Default wird Zugriff auf Lesen gestellt.

Rückgabe:

JSF forward String

Ausnahmebehandlung:

Throwable

Definiert in Zeile 1972 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.metaTypeRight, RoleOverview.role, RoleOverview.selAttr, RoleOverview.selMet, RoleOverview.selObj und RoleOverview.selObjType.

void saveAssignments () throws Throwable

Weisst einer Rolle ein oder mehrere Funktionen zu. Aktualisierung der Datenbank, Zuweisungstabelle und der Listbox.

Ausnahmebehandlung:

Throwable

Definiert in Zeile 2614 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedFunc, RoleOverview.bundles, RoleOverview.filtAssignedRights, RoleOverview.role, RoleOverview.roleId, RoleOverview.selectableRights, RoleOverview.selectedRights und RoleOverview.standardOneMenu.

void saveRights () throws Throwable

Speichert Rechte für die Ausgewählten SelectItems in dem Metamodelle Tab. Gespeichert werden die Items, die zuletzt ausgewählt wurden

Ausnahmebehandlung:

Throwable

Definiert in Zeile 1696 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.ATTRIBUTE_RIGHT,
RoleOverview.convertAccess(), RoleOverview.currentAttr, RoleOverview.currentMet,
RoleOverview.currentObj, RoleOverview.getAttrInfoById(),
RoleOverview.getMetInfoById(), RoleOverview.getObjInfoById(),
RoleOverview.getRelationById(), RoleOverview.METAMODEL_RIGHT,
RoleOverview.metaTypeRight, RoleOverview.resolveMetamodelDependencies(),
RoleOverview.resolveObjectDependencies(), RoleOverview.role, RoleOverview.roleId,
RoleOverview.selAttr, RoleOverview.selMet und RoleOverview.selObj.

String saveRightsByObjType () throws Throwable

Speichert Rechte aus dem Objekttyp Tab

Ausnahmebehandlung:

Throwable

Definiert in Zeile 1800 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.attr,
RoleOverview.ATTRIBUTE_RIGHT, RoleOverview.convertAccess(),
RoleOverview.currentAttrObj, RoleOverview.currentObjByType,
RoleOverview.currentObjTypes, RoleOverview.getObjAttributeInfoById(),
RoleOverview.getObjByTypeInfoById(), RoleOverview.getObjTypeInfoById(),
RoleOverview.metaTypeRight, RoleOverview.OBJECT_RIGHT,
RoleOverview.OBJECT_TYPE_RIGHT, RoleOverview.resolveObjByTypeDependencies(),
RoleOverview.resolveObjecttypeDependencies(), RoleOverview.role, RoleOverview.roleId,
RoleOverview.selAttrObj, RoleOverview.selObjByType und RoleOverview.selObjType.

String saveRightsByRelType () throws Throwable

Speichert Rechte aus dem Relationstyp Tab. Es werden immer die zuletzt ausgewählten Rechte gespeichert.

Rückgabe:

"success", JSF forward String

Ausnahmebehandlung:

Throwable

Definiert in Zeile 1888 der Datei RoleOverview.java.

Benutzt RoleOverview.assignedRights, RoleOverview.attr,
RoleOverview.ATTRIBUTE_RIGHT, RoleOverview.convertAccess(),
RoleOverview.currentAttrRel, RoleOverview.currentRel, RoleOverview.currentRelTypes,
RoleOverview.getRelAttributeInfoById(), RoleOverview.getRelInfoById(),
RoleOverview.getRelTypeInfoById(), RoleOverview.metaTypeRight, RoleOverview.rel,
RoleOverview.RELATION_RIGHT, RoleOverview.RELATION_TYPE_RIGHT,
RoleOverview.resolveRelationtypeDependencies(),
RoleOverview.resolveRelByTypeDependencies(), RoleOverview.role, RoleOverview.roleId,
RoleOverview.selAttrRel, RoleOverview.selRel und RoleOverview.selRelTypes.

void setAttr (ArrayList< AuthInfo > attr)

Definiert in Zeile 1018 der Datei RoleOverview.java.

void setAttrObj (ArrayList< AuthInfo > attrObj)

Definiert in Zeile 1070 der Datei RoleOverview.java.

void setAttrRel (ArrayList< AuthInfo > attrRel)

Definiert in Zeile 1062 der Datei RoleOverview.java.

void setAttrTypes (ArrayList< AuthInfo > attrTypes)

Definiert in Zeile 1022 der Datei RoleOverview.java.

void setBundleList (ArrayList< SelectItem > bundleList)

Definiert in Zeile 2738 der Datei RoleOverview.java.

void setBundleOneList (String bundleOneList)

Definiert in Zeile 2756 der Datei RoleOverview.java.

void setBundleOneMenu (String bundleOneMenu)

Definiert in Zeile 2769 der Datei RoleOverview.java.

void setBundles (ArrayList< Module > bundles)

Definiert in Zeile 2763 der Datei RoleOverview.java.

void setBundleSelected (boolean bundleSelected)

Definiert in Zeile 2794 der Datei RoleOverview.java.

void setColumnNumbers (int columnNumbers)

Definiert in Zeile 2453 der Datei RoleOverview.java.

void setCreateAccessDefault () [private]

Definiert in Zeile 3129 der Datei RoleOverview.java.

Benutzt RoleOverview.createRight.

Wird benutzt von RoleOverview.getAssignedRights().

void setCreateAttrTyp (boolean createAttrTyp)

Definiert in Zeile 3024 der Datei RoleOverview.java.

void setCreateMeta (boolean createMeta)

Definiert in Zeile 2955 der Datei RoleOverview.java.

void setCreateObjType (boolean createObjType)

Definiert in Zeile 2967 der Datei RoleOverview.java.

void setCreateRelType (boolean createRelType)

Definiert in Zeile 2961 der Datei RoleOverview.java.

void setCreateStatus (boolean createTime)

Definiert in Zeile 3058 der Datei RoleOverview.java.

void setCreateTime (boolean createTime)

Definiert in Zeile 3041 der Datei RoleOverview.java.

void setCurrentAttr (String[] currentAttr)

Definiert in Zeile 1098 der Datei RoleOverview.java.

void setCurrentAttrObj (String[] currentAttrObj)

Definiert in Zeile 1162 der Datei RoleOverview.java.

void setCurrentAttrRel (String[] currentAttrRel)

Definiert in Zeile 1154 der Datei RoleOverview.java.

void setCurrentAttrTypes (String[] currentAttrTypes)

Definiert in Zeile 1114 der Datei RoleOverview.java.

void setCurrentMet (String[] currentMet)

Definiert in Zeile 1086 der Datei RoleOverview.java.

void setCurrentObj (String[] currentObj)

Definiert in Zeile 1092 der Datei RoleOverview.java.

void setCurrentObjByType (String[] currentObjByType)

Definiert in Zeile 1146 der Datei RoleOverview.java.

void setCurrentObjTypes (String[] currentObjTypes)

Definiert in Zeile 1106 der Datei RoleOverview.java.

void setCurrentRel (String[] currentRel)

Definiert in Zeile 1138 der Datei RoleOverview.java.

void setCurrentRelTypes (String[] currentRelTypes)

Definiert in Zeile 1130 der Datei RoleOverview.java.

void setDefaultAccess (String[] defaultAccess)

Definiert in Zeile 1122 der Datei RoleOverview.java.

void setFiltAssignedRights (ArrayList< View > filtAssignedRights)

Definiert in Zeile 2788 der Datei RoleOverview.java.

void setGroups (ArrayList< Group > groups)

Parameter:

groups

Definiert in Zeile 714 der Datei RoleOverview.java.

void setHidden (HtmlInputHidden hidden)

Definiert in Zeile 773 der Datei RoleOverview.java.

void setHidden2 (HtmlInputHidden hidden2)

Definiert in Zeile 872 der Datei RoleOverview.java.

void setHidden3 (HtmlInputHidden hidden3)

Definiert in Zeile 880 der Datei RoleOverview.java.

void setHiddenRoleId (HtmlInputHidden hiddenRoleId)

Definiert in Zeile 2478 der Datei RoleOverview.java.

void setHiddenView (HtmlInputHidden hiddenView)

Definiert in Zeile 2806 der Datei RoleOverview.java.

void setMenu (ManageMenu menu)

Definiert in Zeile 2472 der Datei RoleOverview.java.

void setMetaTypeRight (boolean metaTypeRight)

Definiert in Zeile 2444 der Datei RoleOverview.java.

void setNotAssignedGroups (ArrayList< Group > notAssignedGroups)

Parameter:

notAssignedGroups

Definiert in Zeile 864 der Datei RoleOverview.java.

void setObjByType (ArrayList< AuthInfo > objByType)

Definiert in Zeile 1046 der Datei RoleOverview.java.

void setObjects (ArrayList< AuthInfo > objects)

Definiert in Zeile 1014 der Datei RoleOverview.java.

void setObjTypes (ArrayList< AuthInfo > objTypes)

Definiert in Zeile 1006 der Datei RoleOverview.java.

void setRel (ArrayList< AuthInfo > rel)

Definiert in Zeile 1038 der Datei RoleOverview.java.

void setRelations (ArrayList< AuthInfo > relations)

Definiert in Zeile 1054 der Datei RoleOverview.java.

void setRelTypes (ArrayList< AuthInfo > relTypes)

Definiert in Zeile 1030 der Datei RoleOverview.java.

void setRightTypeChoice (String rightTypeChoice)

Definiert in Zeile 2437 der Datei RoleOverview.java.

void setRoleId (int roleId)

Parameter:

roleId

Definiert in Zeile 789 der Datei RoleOverview.java.

void setRoles (ArrayList< Role > roles)

Parameter:

roles

Definiert in Zeile 765 der Datei RoleOverview.java.

void setSelAccess (ArrayList< SelectItem > selAccess)

Definiert in Zeile 1387 der Datei RoleOverview.java.

void setSelAttr (ArrayList< SelectItem > selAttr)

Definiert in Zeile 1311 der Datei RoleOverview.java.

void setSelAttrObj (ArrayList< SelectItem > selAttrObj)

Definiert in Zeile 1559 der Datei RoleOverview.java.

void setSelAttrRel (ArrayList< SelectItem > selAttrRel)

Definiert in Zeile 1523 der Datei RoleOverview.java.

void setSelectableRights (ArrayList< SelectItem > selectableRights)

Definiert in Zeile 2744 der Datei RoleOverview.java.

void setSelectedRights (ArrayList< String > selectedBundleRoles)

Definiert in Zeile 2750 der Datei RoleOverview.java.

Benutzt RoleOverview.selectedRights.

void setSelMet (ArrayList< SelectItem > selMet)

Definiert in Zeile 1208 der Datei RoleOverview.java.

void setSelObj (ArrayList< SelectItem > selObj)

Definiert in Zeile 1263 der Datei RoleOverview.java.

void setSelObjByType (ArrayList< SelectItem > selObjByType)

Definiert in Zeile 1487 der Datei RoleOverview.java.

void setSelObjType (ArrayList< SelectItem > selObjType)

Definiert in Zeile 1354 der Datei RoleOverview.java.

void setSelRel (ArrayList< SelectItem > selRel)

Definiert in Zeile 1423 der Datei RoleOverview.java.

void setSelRelTypes (ArrayList< SelectItem > selRelTypes)

Definiert in Zeile 1452 der Datei RoleOverview.java.

void setSortAscendingOver (boolean sortAscendingOver)

Definiert in Zeile 2466 der Datei RoleOverview.java.

Wird benutzt von ManageMenu.goToRoles().

void setSortColumnOver (String sortColumnOver)

Definiert in Zeile 2460 der Datei RoleOverview.java.

Wird benutzt von ManageMenu.goToRoles().

void setSortQueriesAssignedAscending (boolean sortQueriesAssignedAscending)

Definiert in Zeile 2818 der Datei RoleOverview.java.

void setSortQueriesAssignedColumn (String sortQueriesAssignedColumn)

Definiert in Zeile 2812 der Datei RoleOverview.java.

void setSortQueriesNotAssignedAscending (boolean sortQueriesNotAssignedAscending)

Definiert in Zeile 2830 der Datei RoleOverview.java.

void setSortQueriesNotAssignedColumn (String sortQueriesNotAssignedColumn)

Definiert in Zeile 2824 der Datei RoleOverview.java.

void setStandardOneMenu (String standardOneMenu)

Definiert in Zeile 2781 der Datei RoleOverview.java.

void setStandardRoles (ArrayList< SelectItem > standardRoles)

Definiert in Zeile 2775 der Datei RoleOverview.java.

void setStandardSelected (boolean standardSelected)

Definiert in Zeile 2800 der Datei RoleOverview.java.

void setTableName (String tableName)

Parameter:

tableName

Definiert in Zeile 971 der Datei RoleOverview.java.

void setUsers (ArrayList< User > users)

Parameter:

users

Definiert in Zeile 732 der Datei RoleOverview.java.

void switchDataMeta (ValueChangeEvent event)

Definiert in Zeile 1631 der Datei RoleOverview.java.

Benutzt RoleOverview.listLoaded, RoleOverview.metaTypeRight und RoleOverview.rightTypeChoice.

**void validateName (FacesContext context, UIComponent component, Object value)
throws ValidatorException**

Validiert den Namen der Rolle bei der Erstellung

Parameter:

context
component
value

Ausnahmebehandlung:**ValidatorException**

Definiert in Zeile 678 der Datei RoleOverview.java.

Benutzt RoleOverview.role und RoleOverview.roles.

3.2.3.4 Dokumentation der Datenelemente**ArrayList<AuthInfo> assignedFilteredRights [private]**

Definiert in Zeile 217 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getAssignedFilteredRights() und RoleOverview.RoleOverview().

ArrayList<View> assignedFunc = new ArrayList<View>() [private]

Definiert in Zeile 83 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.alreadyAssigned(), RoleOverview.deleteSelectedFunctions(), RoleOverview.listenerSelStandard(), RoleOverview.loadFuncRights(), RoleOverview.removeOneAssignment() und RoleOverview.saveAssignments().

LinkedList<Query> assignedQueries [private]

Definiert in Zeile 107 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getAssignedQueries(), RoleOverview.getNotAssignedQueries(), RoleOverview.queryAssign(), RoleOverview.queryRemoveOne() und RoleOverview.RoleOverview().

ArrayList<AuthInfo> assignedRights [private]

Definiert in Zeile 216 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.deleteRight(), RoleOverview.deleteSelectedRights(), RoleOverview.getAssignedRights(), RoleOverview.getSelMet(), RoleOverview.getSelObjByType(), RoleOverview.getSelObjType(), RoleOverview.getSelRelTypes(), RoleOverview.isInAssignedRights(), RoleOverview.resolveMetamodelDependencies(), RoleOverview.resolveObjByTypeDependencies(), RoleOverview.resolveObjectDependencies(),

RoleOverview.resolveObjecttypeDependencies(),
RoleOverview.resolveRelationtypeDependencies(),
RoleOverview.resolveRelByTypeDependencies(), RoleOverview.RoleOverview(),
RoleOverview.saveAccess(), RoleOverview.saveRights(),
RoleOverview.saveRightsByObjType() und RoleOverview.saveRightsByRelType().

ArrayList<AuthInfo> attr [private]

Definiert in Zeile 127 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getAttr(), RoleOverview.getAttrInfoById(),
RoleOverview.getSelAttr(), RoleOverview.resolveObjByTypeDependencies(),
RoleOverview.resolveObjectDependencies(),
RoleOverview.resolveObjecttypeDependencies(),
RoleOverview.resolveRelationtypeDependencies(), RoleOverview.RoleOverview(),
RoleOverview.saveRightsByObjType() und RoleOverview.saveRightsByRelType().

final String ATTRIBUTE_RIGHT = IAuthNames.METAMODEL_ATTRIBUTE [private]

Definiert in Zeile 199 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelAttr(), RoleOverview.getSelAttrObj(),
RoleOverview.getSelAttrRel(), RoleOverview.resolveMetamodelDependencies(),
RoleOverview.resolveObjByTypeDependencies(),
RoleOverview.resolveObjecttypeDependencies(),
RoleOverview.resolveRelationtypeDependencies(),
RoleOverview.resolveRelByTypeDependencies(), RoleOverview.saveRights(),
RoleOverview.saveRightsByObjType() und RoleOverview.saveRightsByRelType().

ArrayList<AuthInfo> attrObj [private]

Definiert in Zeile 133 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getAttrObj(), RoleOverview.getObjAttributeInfoById(),
RoleOverview.getSelAttrObj(), RoleOverview.resolveObjByTypeDependencies() und
RoleOverview.RoleOverview().

ArrayList<AuthInfo> attrRel [private]

Definiert in Zeile 140 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getAttrRel(), RoleOverview.getRelAttributeInfoById(),
RoleOverview.getSelAttrRel(), RoleOverview.resolveRelByTypeDependencies() und
RoleOverview.RoleOverview().

ArrayList<AuthInfo> attrTypes [private]

Definiert in Zeile 135 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getAttrTypes() und RoleOverview.RoleOverview().

String bundleId = "0" [private]

Definiert in Zeile 62 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.deleteSelectedFunctions(),
RoleOverview.listenerSelBundle(), RoleOverview.listenerSelStandard(),
RoleOverview.loadFuncRights() und RoleOverview.removeOneAssignment().

ArrayList<SelectItem> bundleList = new ArrayList<SelectItem>() [private]

Definiert in Zeile 59 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getBundleList() und RoleOverview.loadFuncRights().

String bundleOneList = "" [private]

Definiert in Zeile 71 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getBundleOneList() und RoleOverview.loadFuncRights().

String bundleOneMenu = "" [private]

Definiert in Zeile 74 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getBundleOneMenu() und
RoleOverview.loadFuncRights().

ArrayList<Module> bundles = new ArrayList<Module>() [private]

Definiert in Zeile 56 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getBundles(), RoleOverview.listenerSelBundle(),
RoleOverview.listenerSelStandard(), RoleOverview.loadFuncRights() und
RoleOverview.saveAssignments().

boolean bundleSelected = false [private]

Definiert in Zeile 90 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.isBundleSelected(), RoleOverview.listenerSelBundle() und
RoleOverview.loadFuncRights().

int columnNumbers = 2 [private]

Definiert in Zeile 120 der Datei RoleOverview.java.

boolean [] createRight = {false, false, false, false, false, false} [private]

Definiert in Zeile 95 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getAssignedRights(), RoleOverview.isCreateAttrTyp(), RoleOverview.isCreateMeta(), RoleOverview.isCreateObjType(), RoleOverview.isCreateRelType(), RoleOverview.isCreateStatus(), RoleOverview.isCreateTime(), RoleOverview.listenerCreateAttrTyp(), RoleOverview.listenerCreateMeta(), RoleOverview.listenerCreateObjType(), RoleOverview.listenerCreateRelType(), RoleOverview.listenerCreateStatus(), RoleOverview.listenerCreateTime() und RoleOverview.setCreateAccessDefault().

String [] currentAttr [private]

Definiert in Zeile 148 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getCurrentAttr(), RoleOverview.objSelected() und RoleOverview.saveRights().

String [] currentAttrObj [private]

Definiert in Zeile 152 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getCurrentAttrObj() und RoleOverview.saveRightsByObjType().

String [] currentAttrRel [private]

Definiert in Zeile 151 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getCurrentAttrRel() und RoleOverview.saveRightsByRelType().

String [] currentAttrTypes [private]

Definiert in Zeile 147 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getCurrentAttrTypes().

String [] currentMet [private]

Definiert in Zeile 143 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.deleteRight(), RoleOverview.deleteSelectedRights(), RoleOverview.getCurrentMet(), RoleOverview.getSelObj(), RoleOverview.modellSelected() und RoleOverview.saveRights().

String [] currentObj [private]

Definiert in Zeile 145 der Datei RoleOverview.java.

Wird benutzt von `RoleOverview.getCurrentObj()`, `RoleOverview.getSelAttr()`, `RoleOverview.objSelected()` und `RoleOverview.saveRights()`.

String [] currentObjByType [private]

Definiert in Zeile 146 der Datei `RoleOverview.java`.

Wird benutzt von `RoleOverview.getCurrentObjByType()`, `RoleOverview.getSelAttrObj()`, `RoleOverview.objByTypeSelected()` und `RoleOverview.saveRightsByObjType()`.

String [] currentObjTypes [private]

Definiert in Zeile 144 der Datei `RoleOverview.java`.

Wird benutzt von `RoleOverview.deleteRight()`, `RoleOverview.deleteSelectedRights()`, `RoleOverview.getCurrentObjTypes()`, `RoleOverview.getSelObjByType()`, `RoleOverview.objTypeSelected()` und `RoleOverview.saveRightsByObjType()`.

String [] currentRel [private]

Definiert in Zeile 150 der Datei `RoleOverview.java`.

Wird benutzt von `RoleOverview.getCurrentRel()`, `RoleOverview.getSelAttrRel()`, `RoleOverview.relByTypeSelected()` und `RoleOverview.saveRightsByRelType()`.

String [] currentRelTypes [private]

Definiert in Zeile 149 der Datei `RoleOverview.java`.

Wird benutzt von `RoleOverview.deleteRight()`, `RoleOverview.deleteSelectedRights()`, `RoleOverview.getCurrentRelTypes()`, `RoleOverview.getSelRel()`, `RoleOverview.relTypeSelected()` und `RoleOverview.saveRightsByRelType()`.

int DEFAULT_CREATE_RIGHT = 15 [static, private]

Definiert in Zeile 97 der Datei `RoleOverview.java`.

Wird benutzt von `RoleOverview.listenerCreateAttrTyp()`, `RoleOverview.listenerCreateMeta()`, `RoleOverview.listenerCreateObjType()`, `RoleOverview.listenerCreateRelType()`, `RoleOverview.listenerCreateStatus()` und `RoleOverview.listenerCreateTime()`.

String [] defaultAccess [private]

Definiert in Zeile 153 der Datei `RoleOverview.java`.

Wird benutzt von `RoleOverview.convertAccess()`, `RoleOverview.getDefaultAccess()` und `RoleOverview.RoleOverview()`.

ArrayList<View> filtAssignedRights = new ArrayList<View>() [private]

Definiert in Zeile 86 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.deleteSelectedFunctions(), RoleOverview.getFiltAssignedRights(), RoleOverview.listenerSelStandard(), RoleOverview.loadFuncRights(), RoleOverview.removeOneAssignment() und RoleOverview.saveAssignments().

ArrayList<Group> groups [private]

Definiert in Zeile 103 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.addGroup(), RoleOverview.back(), RoleOverview.deleteGroup(), RoleOverview.getGroups(), RoleOverview.getNotAssignedGroups() und RoleOverview.RoleOverview().

HtmlInputHidden hidden [private]

Definiert in Zeile 109 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getHidden(), RoleOverview.goToDeleteFromOverview(), RoleOverview.queryRemoveOne() und RoleOverview.RoleOverview().

HtmlInputHidden hidden2 [private]

Definiert in Zeile 110 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.deleteGroup(), RoleOverview.getHidden2() und RoleOverview.RoleOverview().

HtmlInputHidden hidden3 [private]

Definiert in Zeile 111 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.deleteUser(), RoleOverview.getHidden3() und RoleOverview.RoleOverview().

HtmlInputHidden hiddenRoleId [private]

Definiert in Zeile 112 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getHiddenRoleId() und RoleOverview.RoleOverview().

HtmlInputHidden hiddenView [private]

Definiert in Zeile 88 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getHiddenView() und RoleOverview.removeOneAssignment().

boolean listLoaded = false [private]

Definiert in Zeile 116 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.back(), RoleOverview.createRole(), RoleOverview.editAssignedRole(), RoleOverview.getAssignedRights(), RoleOverview.goToEditRole() und RoleOverview.switchDataMeta().

ManageMenu menu [private]

Definiert in Zeile 117 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.editAssignedRole(), RoleOverview.getMenu(), RoleOverview.getRoles(), RoleOverview.goAssigned(), RoleOverview.goToEditRole() und RoleOverview.RoleOverview().

ArrayList<AuthInfo> meta [private]

Definiert in Zeile 122 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getMeta(), RoleOverview.getMetInfoById(), RoleOverview.getSelMet() und RoleOverview.RoleOverview().

final String METAMODEL_RIGHT = IAuthNames.METAMODEL [private]

Definiert in Zeile 196 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelMet() und RoleOverview.saveRights().

boolean metaSelected = false [private]

Definiert in Zeile 186 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelObj() und RoleOverview.modellSelected().

boolean metaTypeRight = true [private]

Definiert in Zeile 193 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.createRole(), RoleOverview.deleteRight(), RoleOverview.deleteSelectedRights(), RoleOverview.editAssignedRole(), RoleOverview.getAssignedRights(), RoleOverview.getColumnNumbers(), RoleOverview.getTableName(), RoleOverview.goToEditRole(), RoleOverview.isMetaTypeRight(), RoleOverview.resolveMetamodelDependencies(), RoleOverview.resolveObjByTypeDependencies(), RoleOverview.resolveObjectDependencies(), RoleOverview.resolveObjecttypeDependencies(), RoleOverview.resolveRelationtypeDependencies(), RoleOverview.resolveRelByTypeDependencies(), RoleOverview.saveAccess(),

RoleOverview.saveRights(), RoleOverview.saveRightsByObjType(),
RoleOverview.saveRightsByRelType() und RoleOverview.switchDataMeta().

final String noFilter = "nofilter" [private]

Definiert in Zeile 92 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.deleteSelectedFunctions(),
RoleOverview.listenerSelBundle(), RoleOverview.removeOneAssignment() und
RoleOverview.RoleOverview().

ArrayList<Group> notAssignedGroups [private]

Definiert in Zeile 104 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.addGroup(), RoleOverview.deleteGroup(),
RoleOverview.getNotAssignedGroups() und RoleOverview.RoleOverview().

LinkedList<Query> notAssignedQueries [private]

Definiert in Zeile 108 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getNotAssignedQueries(), RoleOverview.queryAssign(),
RoleOverview.queryRemoveOne() und RoleOverview.RoleOverview().

ArrayList<User> notAssignedUsers [private]

Definiert in Zeile 106 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.addUser(), RoleOverview.deleteUser(),
RoleOverview.getNotAssignedUsers() und RoleOverview.RoleOverview().

ArrayList<AuthInfo> objByType [private]

Definiert in Zeile 129 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getObjByType(),
RoleOverview.getObjByTypeInfoById(), RoleOverview.getSelObjByType(),
RoleOverview.resolveObjecttypeDependencies() und RoleOverview.RoleOverview().

boolean objByTypeSelected = false [private]

Definiert in Zeile 192 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelAttrObj() und RoleOverview.objByTypeSelected().

final String OBJECT_RIGHT = IAuthNames.METAMODEL_OBJECT [private]

Definiert in Zeile 198 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelAttr(), RoleOverview.getSelObj(), RoleOverview.getSelObjByType(), RoleOverview.resolveMetamodelDependencies(), RoleOverview.resolveObjecttypeDependencies() und RoleOverview.saveRightsByObjType().

**final String OBJECT_TYPE_RIGHT =
IAuthNames.METAMODEL_OBJECT_TYPE [private]**

Definiert in Zeile 197 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelObjType() und RoleOverview.saveRightsByObjType().

ArrayList<AuthInfo> objects [private]

Definiert in Zeile 124 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getObjects(), RoleOverview.getObjInfoById(), RoleOverview.getSelObj(), RoleOverview.getSelObjByType(), RoleOverview.resolveMetamodelDependencies() und RoleOverview.RoleOverview().

boolean objSelected = false [private]

Definiert in Zeile 188 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelAttr(), RoleOverview.modellSelected() und RoleOverview.objSelected().

ArrayList<AuthInfo> objTypes [private]

Definiert in Zeile 131 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getObjTypeInfoById(), RoleOverview.getObjTypes(), RoleOverview.getSelObjType() und RoleOverview.RoleOverview().

boolean objTypeSelected = false [private]

Definiert in Zeile 187 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelObjByType() und RoleOverview.objTypeSelected().

ArrayList<AuthInfo> rel [private]

Definiert in Zeile 138 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getRel(), RoleOverview.getRelInfoById(), RoleOverview.getSelAttrRel(), RoleOverview.getSelRel(),

RoleOverview.resolveRelationtypeDependencies(), RoleOverview.RoleOverview() und RoleOverview.saveRightsByRelType().

final String RELATION_RIGHT = IAuthNames.METAMODEL_RELATION
[private]

Definiert in Zeile 200 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelObj(), RoleOverview.getSelRel(), RoleOverview.resolveMetamodelDependencies(), RoleOverview.resolveRelationtypeDependencies() und RoleOverview.saveRightsByRelType().

final String RELATION_TYPE_RIGHT =
IAuthNames.METAMODEL_RELATION_TYPE [private]

Definiert in Zeile 201 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelRelTypes() und RoleOverview.saveRightsByRelType().

ArrayList<AuthInfo> relations [private]

Definiert in Zeile 125 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getRelationById(), RoleOverview.getRelations(), RoleOverview.getSelObj(), RoleOverview.getSelRel(), RoleOverview.resolveMetamodelDependencies() und RoleOverview.RoleOverview().

boolean relByTypeSelected = false [private]

Definiert in Zeile 191 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelAttrRel() und RoleOverview.relByTypeSelected().

ArrayList<AuthInfo> relTypes [private]

Definiert in Zeile 136 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getRelTypeInfoById(), RoleOverview.getRelTypes(), RoleOverview.getSelRelTypes() und RoleOverview.RoleOverview().

boolean relTypeSelected = false [private]

Definiert in Zeile 190 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelRel() und RoleOverview.relTypeSelected().

String rightTypeChoice = "true" [package]

Definiert in Zeile 194 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getRightTypeChoice(), RoleOverview.goToEditRole() und RoleOverview.switchDataMeta().

Role role [private]

Definiert in Zeile 114 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.createRole(), RoleOverview.deleteRight(), RoleOverview.deleteRole(), RoleOverview.deleteSelectedFunctions(), RoleOverview.deleteSelectedRights(), RoleOverview.editAssignedRole(), RoleOverview.editRole(), RoleOverview.getRole(), RoleOverview.goToCreateNewRole(), RoleOverview.goToDeleteFromOverview(), RoleOverview.goToEditRole(), RoleOverview.queryAssign(), RoleOverview.queryRemoveOne(), RoleOverview.removeOneAssignment(), RoleOverview.RoleOverview(), RoleOverview.saveAccess(), RoleOverview.saveAssignments(), RoleOverview.saveRights(), RoleOverview.saveRightsByObjType(), RoleOverview.saveRightsByRelType() und RoleOverview.validateName().

int roleId [private]

Definiert in Zeile 113 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.addGroup(), RoleOverview.addUser(), RoleOverview.createRole(), RoleOverview.deleteGroup(), RoleOverview.deleteRight(), RoleOverview.deleteRole(), RoleOverview.deleteSelectedFunctions(), RoleOverview.deleteSelectedRights(), RoleOverview.deleteUser(), RoleOverview.editAssignedRole(), RoleOverview.editRole(), RoleOverview.getAssignedQueries(), RoleOverview.getAssignedRights(), RoleOverview.getGroupsByRole(), RoleOverview.getRoleId(), RoleOverview.getUsersByRole(), RoleOverview.goToDeleteFromOverview(), RoleOverview.goToEditRole(), RoleOverview.listenerCreateAttrTyp(), RoleOverview.listenerCreateMeta(), RoleOverview.listenerCreateObjType(), RoleOverview.listenerCreateRelType(), RoleOverview.listenerCreateStatus(), RoleOverview.listenerCreateTime(), RoleOverview.loadFuncRights(), RoleOverview.queryAssign(), RoleOverview.queryRemoveOne(), RoleOverview.removeOneAssignment(), RoleOverview.resolveMetamodelDependencies(), RoleOverview.resolveObjByTypeDependencies(), RoleOverview.resolveObjectDependencies(), RoleOverview.resolveObjecttypeDependencies(), RoleOverview.resolveRelationtypeDependencies(), RoleOverview.resolveRelByTypeDependencies(), RoleOverview.RoleOverview(), RoleOverview.saveAssignments(), RoleOverview.saveRights(), RoleOverview.saveRightsByObjType() und RoleOverview.saveRightsByRelType().

ArrayList<Role> roles [private]

Definiert in Zeile 102 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.createRole(), RoleOverview.getRoles(), RoleOverview.goToDeleteFromOverview(), RoleOverview.goToEditRole(), RoleOverview.RoleOverview() und RoleOverview.validateName().

ArrayList<SelectedItem> selAccess [private]

Definiert in Zeile 171 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelAccess() und RoleOverview.RoleOverview().

ArrayList<SelectedItem> selAttr [private]

Definiert in Zeile 168 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.deleteRight(), RoleOverview.deleteSelectedRights(), RoleOverview.getSelAttr(), RoleOverview.modellSelected(), RoleOverview.objSelected(), RoleOverview.RoleOverview(), RoleOverview.saveAccess() und RoleOverview.saveRights().

ArrayList<SelectedItem> selAttrObj [private]

Definiert in Zeile 183 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelAttrObj(), RoleOverview.objByTypeSelected(), RoleOverview.RoleOverview() und RoleOverview.saveRightsByObjType().

ArrayList<SelectedItem> selAttrRel [private]

Definiert in Zeile 180 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelAttrRel(), RoleOverview.relByTypeSelected(), RoleOverview.RoleOverview() und RoleOverview.saveRightsByRelType().

ArrayList<SelectedItem> selectableRights = new ArrayList<SelectedItem>() [private]

Definiert in Zeile 65 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.deleteSelectedFunctions(), RoleOverview.getSelectableRights(), RoleOverview.listenerSelBundle(), RoleOverview.listenerSelStandard(), RoleOverview.loadFuncRights(), RoleOverview.removeOneAssignment() und RoleOverview.saveAssignments().

boolean selectAllRights = false [private]

Definiert in Zeile 189 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getAssignedRights().

ArrayList<String> selectedRights = new ArrayList<String>() [private]

Definiert in Zeile 68 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelectedRights(), RoleOverview.saveAssignments() und RoleOverview.setSelectedRights().

ArrayList<SelectItem> selMet [private]

Definiert in Zeile 156 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelMet(), RoleOverview.RoleOverview(), RoleOverview.saveAccess() und RoleOverview.saveRights().

ArrayList<SelectItem> selObj [private]

Definiert in Zeile 162 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.deleteRight(), RoleOverview.deleteSelectedRights(), RoleOverview.getSelObj(), RoleOverview.modellSelected(), RoleOverview.RoleOverview(), RoleOverview.saveAccess() und RoleOverview.saveRights().

ArrayList<SelectItem> selObjByType [private]

Definiert in Zeile 165 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.deleteRight(), RoleOverview.deleteSelectedRights(), RoleOverview.getSelObjByType(), RoleOverview.objTypeSelected(), RoleOverview.RoleOverview() und RoleOverview.saveRightsByObjType().

ArrayList<SelectItem> selObjType [private]

Definiert in Zeile 159 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getSelObjType(), RoleOverview.RoleOverview(), RoleOverview.saveAccess() und RoleOverview.saveRightsByObjType().

ArrayList<SelectItem> selRel [private]

Definiert in Zeile 177 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.deleteRight(), RoleOverview.deleteSelectedRights(), RoleOverview.getSelRel(), RoleOverview.relTypeSelected(), RoleOverview.RoleOverview() und RoleOverview.saveRightsByRelType().

ArrayList<SelectItem> selRelTypes [private]

Definiert in Zeile 174 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.deleteRight(), RoleOverview.getSelRelTypes(), RoleOverview.RoleOverview() und RoleOverview.saveRightsByRelType().

boolean sortAscendingOver = true [private]

Definiert in Zeile 206 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.back(), RoleOverview.deleteRole() und RoleOverview.isSortAscendingOver().

String sortColumnOver = null [private]

Definiert in Zeile 205 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.back(), RoleOverview.deleteRole() und RoleOverview.getSortColumnOver().

boolean sortQueriesAssignedAscending = true [private]

Definiert in Zeile 209 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.editAssignedRole(), RoleOverview.goAssigned(), RoleOverview.goToEditRole() und RoleOverview.isSortQueriesAssignedAscending().

String sortQueriesAssignedColumn = null [private]

Definiert in Zeile 208 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.editAssignedRole(), RoleOverview.getSortQueriesAssignedColumn(), RoleOverview.goAssigned() und RoleOverview.goToEditRole().

boolean sortQueriesNotAssignedAscending = true [private]

Definiert in Zeile 212 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.editAssignedRole(), RoleOverview.goAssigned(), RoleOverview.goToEditRole() und RoleOverview.isSortQueriesNotAssignedAscending().

String sortQueriesNotAssignedColumn = null [private]

Definiert in Zeile 211 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.editAssignedRole(), RoleOverview.getSortQueriesNotAssignedColumn(), RoleOverview.goAssigned() und RoleOverview.goToEditRole().

String standardOneMenu = "" [private]

Definiert in Zeile 80 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.deleteSelectedFunctions(), RoleOverview.getStandardOneMenu(), RoleOverview.listenerSelBundle(), RoleOverview.listenerSelStandard(), RoleOverview.removeOneAssignment(), RoleOverview.RoleOverview() und RoleOverview.saveAssignments().

ArrayList<SelectItem> standardRoles = new ArrayList<SelectItem>() [private]

Definiert in Zeile 77 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getStandardRoles() und RoleOverview.RoleOverview().

boolean standardSelected = false [private]

Definiert in Zeile 91 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.isStandardSelected(), RoleOverview.listenerSelStandard() und RoleOverview.loadFuncRights().

String tableName = "" [private]

Definiert in Zeile 219 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.getTableName().

ArrayList<User> users [private]

Definiert in Zeile 105 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.addUser(), RoleOverview.back(), RoleOverview.deleteUser(), RoleOverview.getNotAssignedUsers(), RoleOverview.getUsers() und RoleOverview.RoleOverview().

boolean visibleLinks [private]

Definiert in Zeile 101 der Datei RoleOverview.java.

Wird benutzt von RoleOverview.createRole(), RoleOverview.editAssignedRole(), RoleOverview.goToEditRole() und RoleOverview.isVisibleLinks().

3.2.4 UserOverview Klassenreferenz

3.2.4.1 Ausführliche Beschreibung

Die Bean fuer die Verwaltung von Benutzern. Bearbeitung von Vorname, Nachname, Loginname, Passwort, Arbeitsplatz Zuweisung von Gruppen und Rollen

Autor:

Christian R.

Definiert in Zeile 32 der Datei UserOverview.java.

Öffentliche Methoden

- void assignGroup () throws Throwable
- void assignRole () throws Throwable
- String back ()
- String deleteUser () throws Throwable
- String editAssignedUser () throws Throwable
- ArrayList< Group > getAssignedGroups () throws Throwable
- HtmlInputHidden getAssignedId ()
- ArrayList< Role > getAssignedRoles () throws Throwable
- HtmlInputHidden getHiddenUserId ()
- ManageMenu getMenu ()
- ArrayList< Group > getNotAssignedGroups ()
- HtmlInputHidden getNotAssignedId ()
- ArrayList< Role > getNotAssignedRoles ()
- String getPasswordCheck ()
- String getSortColumnOver ()
- User getUser ()
- ArrayList< User > getUsers () throws Throwable
- void goAssigned (ActionEvent ae)
- String goToCreateUser ()
- String goToDeleteFromEdit ()
- String goToDeleteFromOverview ()
- String goToEdit ()
- String goToEditAfterCreate () throws Throwable
- boolean isSortAscendingOver ()
- boolean isVisibleLinks ()
- void removeGroupAssignment () throws Throwable
- void removeRoleAssignment () throws Throwable
- void saveUser () throws Throwable
- void setAssignedId (HtmlInputHidden assignedId)
- void setHiddenUserId (HtmlInputHidden hiddenUserId)
- void setMenu (ManageMenu menu)
- void setNotAssignedId (HtmlInputHidden notAssignedId)

- void setPasswordCheck (String passwordCheck)
- void setSortAscendingOver (boolean sortAscendingOver)
- void setSortColumnOver (String sortColumnOver)
- void setVisibleLinks (boolean visibleLinks)
- UserOverview ()
- void validateLogin (FacesContext context, UIComponent component, Object value)
throws ValidatorException

Private Methoden

- String encryptPassword (String password)

Private Attribute

- ArrayList< Group > assignedGroups
- HtmlInputHidden assignedId
- ArrayList< Role > assignedRoles
- HtmlInputHidden hiddenUserId
- ManageMenu menu
- ArrayList< Group > notAssignedGroups
- HtmlInputHidden notAssignedId
- ArrayList< Role > notAssignedRoles
- String passwordCheck
- boolean sortAscendingOver = true
- String sortColumnOver = null
- User user
- int userId
- ArrayList< User > users
- boolean visibleLinks

3.2.4.2 Beschreibung der Konstruktoren und Destruktoren

UserOverview ()

Definiert in Zeile 51 der Datei UserOverview.java.

Benutzt UserOverview.assignedGroups, UserOverview.assignedRoles, UserOverview.hiddenUserId, UserOverview.menu, UserOverview.notAssignedGroups, UserOverview.notAssignedRoles, UserOverview.passwordCheck, UserOverview.sortAscendingOver, UserOverview.sortColumnOver, UserOverview.user, UserOverview.users und UserOverview.visibleLinks.

3.2.4.3 Dokumentation der Elementfunktionen

void assignGroup () throws Throwable

Verknuepft einen Benutzer mit den selektierten Gruppen. Setzt eine neue Faces-Nachricht. Die selektierten Gruppen werden zudem aus der Liste der nicht-zugewiesenen Gruppen entfernt und den zugewiesenen hinzugefuegt.

Ausnahmebehandlung:

Throwable

Definiert in Zeile 250 der Datei UserOverview.java.

Benutzt UserOverview.assignedGroups, UserOverview.notAssignedGroups und UserOverview.userId.

void assignRole () throws Throwable

Verknuepft einen Benutzer mit den selektierten Rollen. Setzt eine neue Faces-Nachricht. Die selektierten Rollen werden zudem aus der Liste der nicht-zugewiesenen Rollen entfernt und den zugewiesenen hinzugefuegt.

Ausnahmebehandlung:

Throwable

Definiert in Zeile 335 der Datei UserOverview.java.

Benutzt UserOverview.assignedRoles, UserOverview.notAssignedRoles und UserOverview.userId.

String back ()

Leitet zu der aufrufenden Instanz zurueck. Im zweifelsfall, also falls keine Instanz gespeichert wurde, wird auf die Uebersichtsseite zurueckgeleitet.

Rückgabe:

String fuer die beschriebene Navigation.

Definiert in Zeile 400 der Datei UserOverview.java.

Benutzt UserOverview.sortAscendingOver und UserOverview.sortColumnOver.

String deleteUser () throws Throwable

Loescht einen Benutzer und seine Verknuepfungen unwiederuflich aus der Datenbank. Setzt die Faces-Nachricht, dass der Benutzer geloescht wurde.

Rückgabe:

String fuer die Navigation mit faces-config auf die Benutzer Uebersichtsseite.

Ausnahmebehandlung:

Throwable

Definiert in Zeile 177 der Datei UserOverview.java.

Benutzt UserOverview.sortAscendingOver, UserOverview.sortColumnOver und UserOverview.userId.

String editAssignedUser () throws Throwable

Aufruf von einer Gruppen oder Rollen Detailseite aus.

Rückgabe:

success fuer faces-config

Ausnahmebehandlung:

Throwable

Definiert in Zeile 417 der Datei UserOverview.java.

Benutzt UserOverview.menu, ManageMenu.setSortAscendingAs(), ManageMenu.setSortAscendingAs2(), ManageMenu.setSortAscendingNoAs(), ManageMenu.setSortAscendingNoAs2(), ManageMenu.setSortColumnAs(), ManageMenu.setSortColumnAs2(), ManageMenu.setSortColumnNoAs(), ManageMenu.setSortColumnNoAs2(), UserOverview.user und UserOverview.visibleLinks.

String encryptPassword (String password) [private]

Verschlüsselt ein Passwort mit MD5.

Parameter:

password,nicht verschlüsselt.

Rückgabe:

das mit MD5 verschlüsselte Passwort.

Definiert in Zeile 447 der Datei UserOverview.java.

Wird benutzt von UserOverview.goToEditAfterCreate().

ArrayList<Group> getAssignedGroups () throws Throwable

Laedt die verknuepften Gruppen aus der Datenbank und filtert die nichtverknuepften aus der Liste der Gruppen. Es werden hier also beide Listen, die verknuepften und auch nicht-verknuepften gesetzt.

Rückgabe:

die verknuepften Gruppen.

Ausnahmebehandlung:**Throwable**

Definiert in Zeile 558 der Datei UserOverview.java.

Benutzt UserOverview.assignedGroups, UserOverview.notAssignedGroups und UserOverview.userId.

HtmlInputHidden getAssignedId ()

Definiert in Zeile 623 der Datei UserOverview.java.

Benutzt UserOverview.assignedId.

ArrayList<Role> getAssignedRoles () throws Throwable

Laedt die verknuepften Rollen aus der Datenbank und filtert die nichtverknuepften aus der Liste der Rollen. Es werden hier also beide Listen, die verknuepften und auch nicht-verknuepften gesetzt.

Rückgabe:

die zugewiesenen Rollen.

Ausnahmebehandlung:**Throwable**

Definiert in Zeile 595 der Datei UserOverview.java.

Benutzt UserOverview.assignedRoles, UserOverview.notAssignedRoles und UserOverview.userId.

HtmlInputHidden getHiddenUserId ()

Definiert in Zeile 527 der Datei UserOverview.java.

Benutzt UserOverview.hiddenUserId.

ManageMenu getMenu ()

Definiert in Zeile 509 der Datei UserOverview.java.

Benutzt UserOverview.menu.

ArrayList<Group> getNotAssignedGroups ()

Definiert in Zeile 584 der Datei UserOverview.java.

Benutzt UserOverview.notAssignedGroups.

HtmlInputHidden getNotAssignedId ()

Definiert in Zeile 629 der Datei UserOverview.java.

Benutzt UserOverview.notAssignedId.

ArrayList<Role> getNotAssignedRoles ()

Definiert in Zeile 620 der Datei UserOverview.java.

Benutzt UserOverview.notAssignedRoles.

String getPasswordCheck ()

Definiert in Zeile 635 der Datei UserOverview.java.

Benutzt UserOverview.passwordCheck.

String getSortColumnOver ()

Definiert in Zeile 515 der Datei UserOverview.java.

Benutzt UserOverview.sortColumnOver.

User getUser ()

Definiert in Zeile 539 der Datei UserOverview.java.

Benutzt UserOverview.user.

ArrayList<User> getUsers () throws Throwable

Definiert in Zeile 542 der Datei UserOverview.java.

Benutzt UserOverview.users.

void goAssigned (ActionEvent ae)

Ist notwendig fuer die Navigation und insbesondere der Rueckfuehrung zwischen den Beans.

Parameter:

ae,ein(ActionEvent, welchem aber keine Bedeutung geschenkt wird.

Definiert in Zeile 505 der Datei UserOverview.java.

String goCreateUser ()

Erstellt ein neues Benutzer-Objekt.

Rückgabe:

String fuer die Navigation mit faces-config auf die Benutzer-Erstellen Seite.

Definiert in Zeile 70 der Datei UserOverview.java.

Benutzt UserOverview.passwordCheck und UserOverview.user.

String goToDeleteFromEdit ()

Leitet den Benutzer weiter zur Seite, die ein Loeschen eines Benutzers ermoeeglicht.

Ursprung, also die Edit-Seite, wird vermerkt, um durch den zurueck-Button auch wieder dort hinzukommen.

Rückgabe:

String fuer die Navigation mit faces-config auf die Bestaetigungseite zum Loeschen.

Definiert in Zeile 146 der Datei UserOverview.java.

String goToDeleteFromOverview ()

Leitet den Benutzer zur Seite, die ein Loeschen eines Benutzers ermoeeglicht. Dabei wird aus der Datatable innerhalb von Overview die Id des angewaehlten Benutzers ausgelesen und die Informationen ueber diesen Benutzer der Loeschen Seite zur Verfuegung gestellt.

Rückgabe:

String fuer die Navigation mit faces-config auf die Bestaetigungseite zum Loeschen.

Definiert in Zeile 158 der Datei UserOverview.java.

Benutzt UserOverview.hiddenUserId, UserOverview.user, UserOverview.userId und UserOverview.users.

String goToEdit ()

Liest aus der Overviewseite uebergebenen Id-Wert von dem angewaehlten Benutzer aus.

Aus der Liste mit den Nutzern wird dann der entsprechende gesucht und zur weiteren Bearbeitung zur Verfuegung gestellt.u

Rückgabe:

String fuer die Navigation mit faces-config auf die Bearbeitungsseite.

Ausnahmebehandlung:**SQLException****DatabaseException**

Definiert in Zeile 123 der Datei UserOverview.java.

Benutzt UserOverview.hiddenUserId, UserOverview.user, UserOverview.userId, UserOverview.users und UserOverview.visibleLinks.

String goToEditAfterCreate () throws Throwable

Speichert den neuen Benutzer in der Datenbank. Setzt die Faces-Nachricht, dass ein neuer Benutzer erstellt wurde. Gibt einen String zurueck, der mit der faces-config dafuer sorgt, dass man zur Edit-Seite weitergeleitet wird.

Rückgabe:

String fuer die faces-config Navigation auf die Bearbeitungsseite.

Ausnahmebehandlung:

Throwable

Definiert in Zeile 84 der Datei UserOverview.java.

Benutzt UserOverview.encryptPassword(), UserOverview.passwordCheck, UserOverview.user, UserOverview.userId und UserOverview.visibleLinks.

boolean isSortAscendingOver ()

Definiert in Zeile 521 der Datei UserOverview.java.

Benutzt UserOverview.sortAscendingOver.

boolean isVisibleLinks ()

Definiert in Zeile 533 der Datei UserOverview.java.

Benutzt UserOverview.visibleLinks.

void removeGroupAssignment () throws Throwable

Entfernt eine Verknuepfung zwischen einer Gruppe und einem Benutzer. Setzt die entsprechende Faces-Nachricht.

Ausnahmebehandlung:

Throwable

Definiert in Zeile 282 der Datei UserOverview.java.

Benutzt UserOverview.assignedGroups, UserOverview.assignedId, UserOverview.notAssignedGroups und UserOverview.userId.

void removeRoleAssignment () throws Throwable

Entfernt eine Verknuepfung zwischen einer Rolle und einem Benutzer. Setzt die entsprechende Faces-Nachricht.

Ausnahmebehandlung:

Throwable

Definiert in Zeile 369 der Datei UserOverview.java.

Benutzt `UserOverview.assignedRoles`, `UserOverview.notAssignedId`, `UserOverview.notAssignedRoles` und `UserOverview.userId`.

`void saveUser () throws Throwable`

Uebertraegt die editierten Werte in die Datenbank und benachrichtigt den Benutzer, dass die Werte uebernommen wurden.

Ausnahmebehandlung:

Throwable

Definiert in Zeile 201 der Datei `UserOverview.java`.

Benutzt `UserOverview.passwordCheck`, `UserOverview.user` und `UserOverview.userId`.

`void setAssignedId (HtmlInputHidden assignedId)`

Definiert in Zeile 626 der Datei `UserOverview.java`.

`void setHiddenUserId (HtmlInputHidden hiddenUserId)`

Definiert in Zeile 530 der Datei `UserOverview.java`.

`void setMenu (ManageMenu menu)`

Definiert in Zeile 512 der Datei `UserOverview.java`.

`void setNotAssignedId (HtmlInputHidden notAssignedId)`

Definiert in Zeile 632 der Datei `UserOverview.java`.

`void setPasswordCheck (String passwordCheck)`

Definiert in Zeile 638 der Datei `UserOverview.java`.

`void setSortAscendingOver (boolean sortAscendingOver)`

Definiert in Zeile 524 der Datei `UserOverview.java`.

Wird benutzt von `ManageMenu.goToUsers()`.

`void setSortColumnOver (String sortColumnOver)`

Definiert in Zeile 518 der Datei `UserOverview.java`.

Wird benutzt von `ManageMenu.goToUsers()`.

void setVisibleLinks (boolean visibleLinks)

Definiert in Zeile 536 der Datei UserOverview.java.

**void validateLogin (FacesContext context, UIComponent component, Object value)
throws ValidatorException**

Validiert den im Login-Feld eingegebenen Wert, indem ueberprueft wird, ob dieser noch zu vergeben ist.

Parameter:

context,der JSF Kontext.

component, die UI-Komponente

value,der Wert der eingegeben wurde.

Ausnahmebehandlung:

ValidatorException, falls der Login schon existiert.

Definiert in Zeile 475 der Datei UserOverview.java.

Benutzt UserOverview.user und UserOverview.users.

3.2.4.4 Dokumentation der Datenelemente

ArrayList<Group> assignedGroups [private]

Definiert in Zeile 36 der Datei UserOverview.java.

Wird benutzt von UserOverview.assignGroup(), UserOverview.getAssignedGroups(), UserOverview.removeGroupAssignment() und UserOverview.UserOverview().

HtmlInputHidden assignedId [private]

Definiert in Zeile 43 der Datei UserOverview.java.

Wird benutzt von UserOverview.getAssignedId() und UserOverview.removeGroupAssignment().

ArrayList<Role> assignedRoles [private]

Definiert in Zeile 38 der Datei UserOverview.java.

Wird benutzt von UserOverview.assignRole(), UserOverview.getAssignedRoles(), UserOverview.removeRoleAssignment() und UserOverview.UserOverview().

HtmlInputHidden hiddenUserId [private]

Definiert in Zeile 42 der Datei UserOverview.java.

Wird benutzt von `UserOverview.getHiddenUserId()`,
`UserOverview.goToDeleteFromOverview()`, `UserOverview.goToEdit()` und
`UserOverview.UserOverview()`.

ManageMenu menu [private]

Definiert in Zeile 48 der Datei `UserOverview.java`.

Wird benutzt von `UserOverview.editAssignedUser()`, `UserOverview.getMenu()` und
`UserOverview.UserOverview()`.

ArrayList<Group> notAssignedGroups [private]

Definiert in Zeile 37 der Datei `UserOverview.java`.

Wird benutzt von `UserOverview.assignGroup()`, `UserOverview.getAssignedGroups()`,
`UserOverview.getNotAssignedGroups()`, `UserOverview.removeGroupAssignment()` und
`UserOverview.UserOverview()`.

HtmlInputHidden notAssignedId [private]

Definiert in Zeile 44 der Datei `UserOverview.java`.

Wird benutzt von `UserOverview.getNotAssignedId()` und
`UserOverview.removeRoleAssignment()`.

ArrayList<Role> notAssignedRoles [private]

Definiert in Zeile 39 der Datei `UserOverview.java`.

Wird benutzt von `UserOverview.assignRole()`, `UserOverview.getAssignedRoles()`,
`UserOverview.getNotAssignedRoles()`, `UserOverview.removeRoleAssignment()` und
`UserOverview.UserOverview()`.

String passwordCheck [private]

Definiert in Zeile 49 der Datei `UserOverview.java`.

Wird benutzt von `UserOverview.getPasswordCheck()`, `UserOverview.goToCreateUser()`,
`UserOverview.goToEditAfterCreate()`, `UserOverview.saveUser()` und
`UserOverview.UserOverview()`.

boolean sortAscendingOver = true [private]

Definiert in Zeile 46 der Datei `UserOverview.java`.

Wird benutzt von `UserOverview.back()`, `UserOverview.deleteUser()`,
`UserOverview.isSortAscendingOver()` und `UserOverview.UserOverview()`.

String sortColumnOver = null [private]

Definiert in Zeile 45 der Datei UserOverview.java.

Wird benutzt von UserOverview.back(), UserOverview.deleteUser(), UserOverview.getSortColumnOver() und UserOverview.UserOverview().

User user [private]

Definiert in Zeile 34 der Datei UserOverview.java.

Wird benutzt von UserOverview.editAssignedUser(), UserOverview.getUser(), UserOverview.goToCreateUser(), UserOverview.goToDeleteFromOverview(), UserOverview.goToEdit(), UserOverview.goToEditAfterCreate(), UserOverview.saveUser(), UserOverview.UserOverview() und UserOverview.validateLogin().

int userId [private]

Definiert in Zeile 41 der Datei UserOverview.java.

Wird benutzt von UserOverview.assignGroup(), UserOverview.assignRole(), UserOverview.deleteUser(), UserOverview.getAssignedGroups(), UserOverview.getAssignedRoles(), UserOverview.goToDeleteFromOverview(), UserOverview.goToEdit(), UserOverview.goToEditAfterCreate(), UserOverview.removeGroupAssignment(), UserOverview.removeRoleAssignment() und UserOverview.saveUser().

ArrayList<User> users [private]

Definiert in Zeile 35 der Datei UserOverview.java.

Wird benutzt von UserOverview.getUsers(), UserOverview.goToDeleteFromOverview(), UserOverview.goToEdit(), UserOverview.UserOverview() und UserOverview.validateLogin().

boolean visibleLinks [private]

Definiert in Zeile 47 der Datei UserOverview.java.

Wird benutzt von UserOverview.editAssignedUser(), UserOverview.goToEdit(), UserOverview.goToEditAfterCreate(), UserOverview.isVisibleLinks() und UserOverview.UserOverview().

3.3 Sonstiges

3.3.1 ExceptionActionListener Klassenreferenz

Abgeleitet von `com::sun::faces::application::ActionListenerImpl` und `javax::faces::event::ActionListener`.

3.3.1.1 Ausführliche Beschreibung

Durch diese Klasse werden alle Actions von dem FacesServlet abgefangen und abgearbeitet. Wird irgendwo eine Exception geworfen, so wird sie hier abgefangen und es wird an eine error Seite weitergeleitet

Autor:

Igor

Definiert in Zeile 21 der Datei `ExceptionActionListener.java`.

Öffentliche Methoden

- `void processAction (ActionEvent event)`

3.3.1.2 Dokumentation der Elementfunktionen

`void processAction (ActionEvent event)`

Definiert in Zeile 24 der Datei `ExceptionActionListener.java`.

3.3.2 File Klassenreferenz

3.3.2.1 Ausführliche Beschreibung

Erlaubt einfaches Lesen und Schreiben von Dateien.

Autor:

Roland Koppe

Definiert in Zeile 17 der Datei `File.java`.

Öffentliche, statische Methoden

- `static String read (String filename)`
- `static boolean write (String filename, boolean append, String write)`

3.3.2.2 Dokumentation der Elementfunktionen

static String read (String filename) [static]

Definiert in Zeile 19 der Datei File.java.

static boolean write (String filename, boolean append, String write) [static]

Definiert in Zeile 48 der Datei File.java.

4 de.offis.pg.eam.core_datamgr

4.1 BundleConfiguration

4.1.1 BundleViews Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::module::api::IModuleView`.

4.1.1.1 Ausführliche Beschreibung

Definiert in Zeile 21 der Datei `BundleViews.java`.

Öffentliche Methoden

- `String getBundleVersion ()`
- `String getDescription ()`
- `long getEquinoxBundleId ()`
- `String getName ()`
- `List< View > getViews ()`

Statische private Attribute

- `static final String VIEWS = "bundle_views.xml"`

4.1.1.2 Dokumentation der Elementfunktionen

String getBundleVersion ()

Definiert in Zeile 83 der Datei `BundleViews.java`.

String getDescription ()

Definiert in Zeile 28 der Datei `BundleViews.java`.

long getEquinoxBundleId ()

Definiert in Zeile 34 der Datei `BundleViews.java`.

String getName ()

Definiert in Zeile 40 der Datei BundleViews.java.

List<View> getViews ()

Definiert in Zeile 46 der Datei BundleViews.java.

Benutzt BundleViews.VIEWS.

4.1.1.3 Dokumentation der Datenelemente

final String VIEWS = "bundle_views.xml" [static, private]

Definiert in Zeile 24 der Datei BundleViews.java.

Wird benutzt von BundleViews.getViews().

4.1.2 Configuration Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::configuration::api::IBundleConfiguration`.

4.1.2.1 Ausführliche Beschreibung

Definiert in Zeile 10 der Datei Configuration.java.

Öffentliche Methoden

- `boolean bundleHasObjects ()`
- `boolean bundleHasRelations ()`
- `boolean bundleHasViews ()`
- `Document getObjectsJdomDocument ()` throws `JDOMException`, `IOException`

4.1.2.2 Dokumentation der Elementfunktionen

boolean bundleHasObjects ()

Definiert in Zeile 13 der Datei Configuration.java.

boolean bundleHasRelations ()

Definiert in Zeile 28 der Datei Configuration.java.

boolean bundleHasViews ()

Definiert in Zeile 18 der Datei Configuration.java.

Document `getObjectsJdomDocument ()` throws `JDOMException`, `IOException`

Definiert in Zeile 23 der Datei `Configuration.java`.

4.1.3 Activator Klassenreferenz

Abgeleitet von `org::osgi::framework::BundleActivator`.

4.1.3.1 Ausführliche Beschreibung

Definiert in Zeile 34 der Datei `Activator.java`.

Öffentliche Methoden

- `void start (BundleContext context) throws Exception`
- `void stop (BundleContext context) throws Exception`

Statische öffentliche Attribute

- `static String bundleSymbolicName`
- `static String bundleVersion`
- `static long equinoxBundleId`

Private Attribute

- `ServiceTracker httpServiceTracker`

Klassen

- `class HttpServiceTracker`
- `class ServletContextListenerServletAdaptor`

4.1.3.2 Dokumentation der Elementfunktionen**`void start (BundleContext context) throws Exception`**

Definiert in Zeile 42 der Datei `Activator.java`.

Benutzt `Activator.bundleSymbolicName`, `Activator.bundleVersion`, `Activator.equinoxBundleId` und `Activator.httpServiceTracker`.

void stop (BundleContext context) throws Exception

Definiert in Zeile 53 der Datei Activator.java.

Benutzt Activator.httpServiceTracker.

4.1.3.3 Dokumentation der Datenelemente

String bundleSymbolicName [static]

Definiert in Zeile 38 der Datei Activator.java.

Wird benutzt von Activator.start().

String bundleVersion [static]

Definiert in Zeile 39 der Datei Activator.java.

Wird benutzt von Activator.start().

long equinoxBundleId [static]

Definiert in Zeile 40 der Datei Activator.java.

Wird benutzt von Activator.start().

ServiceTracker httpServiceTracker [private]

Definiert in Zeile 36 der Datei Activator.java.

Wird benutzt von Activator.start() und Activator.stop().

4.1.4 Activator.HttpServiceTracker Klassenreferenz

Abgeleitet von org::osgi::util::tracker::ServiceTracker.

4.1.4.1 Ausführliche Beschreibung

Definiert in Zeile 58 der Datei Activator.java.

Öffentliche Methoden

- Object addingService (ServiceReference reference)
- HttpServiceTracker (BundleContext context)
- void removedService (ServiceReference reference, Object service)

Statische private Attribute

- static final String PATH = "/datamgr"
- static final String WEB_ROOT = "/WebRoot"

4.1.4.2 Beschreibung der Konstruktoren und Destruktoren

HttpServiceTracker (BundleContext context)

Definiert in Zeile 63 der Datei Activator.java.

4.1.4.3 Dokumentation der Elementfunktionen

Object addingService (ServiceReference reference)

Definiert in Zeile 67 der Datei Activator.java.

Benutzt Activator.HttpServiceTracker.PATH und Activator.HttpServiceTracker.WEB_ROOT.

void removedService (ServiceReference reference, Object service)

Definiert in Zeile 106 der Datei Activator.java.

Benutzt Activator.HttpServiceTracker.PATH.

4.1.4.4 Dokumentation der Datenelemente

final String PATH = "/datamgr" [static, private]

Definiert in Zeile 60 der Datei Activator.java.

Wird benutzt von Activator.HttpServiceTracker.addingService() und Activator.HttpServiceTracker.removedService().

final String WEB_ROOT = "/WebRoot" [static, private]

Definiert in Zeile 61 der Datei Activator.java.

Wird benutzt von Activator.HttpServiceTracker.addingService().

4.1.5 Activator.ServletContextListenerServletAdaptor Klassenreferenz

Abgeleitet von javax::servlet::Servlet.

4.1.5.1 Ausführliche Beschreibung

Definiert in Zeile 115 der Datei Activator.java.

Öffentliche Methoden

- void destroy ()
- ServletConfig getServletConfig ()
- String getServletInfo ()
- void init (ServletConfig config) throws ServletException
- void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException
- ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Private Attribute

- ServletConfig config
- Servlet delegate
- ClassLoader jspLoader
- ServletContextListener listener

4.1.5.2 Beschreibung der Konstruktoren und Destruktoren

ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Definiert in Zeile 121 der Datei Activator.java.

4.1.5.3 Dokumentation der Elementfunktionen

void destroy ()

Definiert in Zeile 155 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config, Activator.ServletContextListenerServletAdaptor.delegate, Activator.ServletContextListenerServletAdaptor.jspLoader und Activator.ServletContextListenerServletAdaptor.listener.

ServletConfig getServletConfig ()

Definiert in Zeile 169 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config.

String getServletInfo ()

Definiert in Zeile 173 der Datei Activator.java.

void init (ServletConfig config) throws ServletException

Definiert in Zeile 129 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config,
Activator.ServletContextListenerServletAdaptor.delegate,
Activator.ServletContextListenerServletAdaptor.jspLoader und
Activator.ServletContextListenerServletAdaptor.listener.

void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException

Definiert in Zeile 143 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.delegate und
Activator.ServletContextListenerServletAdaptor.jspLoader.

4.1.5.4 Dokumentation der Datenelemente**ServletConfig config [private]**

Definiert in Zeile 116 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.getServletConfig() und
Activator.ServletContextListenerServletAdaptor.init().

Servlet delegate [private]

Definiert in Zeile 118 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.init() und
Activator.ServletContextListenerServletAdaptor.service().

ClassLoader jspLoader [private]

Definiert in Zeile 119 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.init() und
Activator.ServletContextListenerServletAdaptor.service().

ServletContextListener listener [private]

Definiert in Zeile 117 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy() und
Activator.ServletContextListenerServletAdaptor.init().

4.1.6 DataMgrContributor Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::api::AbstractAdoptableMenuContributor.

4.1.6.1 Ausführliche Beschreibung

Definiert in Zeile 6 der Datei DataMgrContributor.java.

Öffentliche Methoden

- DataMgrContributor ()
- String getAdopterId ()
- String getContributionTarget ()
- String getContributorId ()
- String getTargetSubmenu ()

4.1.6.2 Beschreibung der Konstruktoren und Destruktoren

DataMgrContributor ()

Definiert in Zeile 8 der Datei DataMgrContributor.java.

4.1.6.3 Dokumentation der Elementfunktionen

String getAdopterId ()

Definiert in Zeile 12 der Datei DataMgrContributor.java.

String getContributionTarget ()

Definiert in Zeile 16 der Datei DataMgrContributor.java.

String getContributorId ()

Definiert in Zeile 20 der Datei DataMgrContributor.java.

String getTargetSubmenu ()

Definiert in Zeile 24 der Datei DataMgrContributor.java.

4.2 Instanzeneingabe

4.2.1 DataObjectManagedBean Klassenreferenz

4.2.1.1 Ausführliche Beschreibung

Diese Klasse enthält die benötigten Methoden für die Instanzeneingabe

Autor:

David Heimann

Roland Koppe

Definiert in Zeile 56 der Datei DataObjectManagedBean.java.

Öffentliche Methoden

- void addFirstMemberToRelation ()
- void addSecondMemberToRelation ()
- void cancelDeleteAll ()
- String cancelPressed ()
- void copyAll ()
- void copyOneInstance ()
- String deleteAll ()
- void deleteAllSelected ()
- String getCheckDelete ()
- EAMCategory getCurrentCategory ()
- List< EAMObject > getCurrentObjects ()
- List< EAMRelation > getCurrentRelations ()
- String getEditDataObject ()
- RelationDataObjectBean getFirstMember ()
- String getFirstMemberDataObjects ()
- String getFirstMemberMultiplicity ()
- EAMSuperClass getMetaObject ()
- EAMObjectDAO getObjectDAOInterface ()
- javax.faces.component.html.HtmlInputHidden getRowCounter ()
- javax.faces.component.html.HtmlInputHidden getRowCounter2 ()
- RelationDataObjectBean getSecondMember ()
- String getSecondMemberDataObjects ()
- String getSecondMemberMultiplicity ()
- boolean isDeleteAll ()
- boolean isInstanceAvailable ()

- boolean isInstanceCopied ()
- boolean isInstancesCopied ()
- String newDataObject ()
- String openCategory ()
- String pasteAll ()
- void pasteInstance ()
- String saveDataObject ()
- void setCurrentCategory (EAMCategory currentCategory)
- void setDeleteAll (boolean deleteAll)
- void setFirstMember (RelationDataObjectBean firstMember)
- void setFirstMemberMultiplicity (String firstMemberMultiplicity)
- void setInstanceAvailable (boolean instanceAvailable)
- void setInstanceCopied (boolean instanceCopied)
- void setInstancesCopied (boolean instancesCopied)
- void setMetaObject (EAMSuperClass metaObject)
- void setObjectDAOInterface (EAMObjectDAO objectDAOInterface)
- void setRowCounter (javax.faces.component.html.HtmlInputHidden rowCounter)
- void setRowCounter2 (javax.faces.component.html.HtmlInputHidden rowCounter2)
- void setSecondMember (RelationDataObjectBean secondMember)
- void setSecondMemberMultiplicity (String secondMemberMultiplicity)
- String showDataObjectFromObject ()
- String showDataObjectFromRelation ()

Öffentliche Attribute

- DataObject dataObject
- DataObjectList dataObjectList = null
- DataObject editObject
- EAMSuperClass metaObject
- EAMObjectDAO objectDAOInterface = DAOFactory.getAuthInstance(userId, password).getEAMObjectDAO()
- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Private Attribute

- String checkDelete

- EAMCategory currentCategory
- List< EAMObject > currentObjects
- List< EAMRelation > currentRelations
- boolean deleteAll = false
- boolean edit
- String editDataObject
- EAMStaticClass factory = EAMStaticClass.getInstance(userId, password, this.getClass().getName())
- RelationDataObjectBean firstMember
- String firstMemberDataObjects
- String firstMemberMultiplicity
- boolean instanceAvailable = true
- boolean instanceCopied = false
- DataObject instanceOfEAMObjectToCopy
- DataObject instanceOfEAMRelationToCopy
- boolean instancesCopied = false
- String instancesOfEAMObjectToCopy
- String instancesOfEAMRelationToCopy
- javax.faces.component.html.HtmlInputHidden rowCounter
- javax.faces.component.html.HtmlInputHidden rowCounter2
- RelationDataObjectBean secondMember
- String secondMemberMultiplicity

Statische private Attribute

- static Lock instanceLock = null

4.2.1.2 Dokumentation der Elementfunktionen

void addFirstMemberToRelation ()

Fügt den ersten RelationMember in das RelationMemberObjekt hinzu, sodass er in der JSP-Seite angezeigt wird.

Definiert in Zeile 526 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.dataObject, DataObjectManagedBean.firstMember, RelationDataObjectBean.setId() und RelationDataObjectBean.setName().

Wird benutzt von DataObjectManagedBean.getFirstMemberDataObjects().

void addSecondMemberToRelation ()

Fügt den zweiten RelationMember in das RelationMemberObjekt hinzu, sodass er in der JSP-Seite angezeigt wird.

Definiert in Zeile 552 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.dataObject, DataObjectManagedBean.secondMember, RelationDataObjectBean.setId() und RelationDataObjectBean.setName().

Wird benutzt von DataObjectManagedBean.getSecondMemberDataObjects().

void cancelDeleteAll ()

Setzt den boolean-Wert deleteAll auf false, sofern im Warnhinweis ein löschen abgelehnt wurde. Somit wird dieser Warnhinweis ausgeblendet.

Definiert in Zeile 189 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.deleteAll().

String cancelPressed ()

Sofern auf "Zurück" geklickt wird, werden die gesetzten Variablen zurückgesetzt.

Definiert in Zeile 1047 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.editObject, DataObjectManagedBean.firstMember, DataObjectManagedBean.instanceLock, DataObjectManagedBean.metaObject, DataObjectManagedBean.secondMember, DataObjectManagedBean.showDataObjectFromObject() und DataObjectManagedBean.showDataObjectFromRelation().

void copyAll ()

Speichert die Id des Metaobjektes in die jeweilige Integervariable, damit sie kopiert werden kann.

Definiert in Zeile 199 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.instancesCopied, DataObjectManagedBean.instancesOfEAMObjectToCopy, DataObjectManagedBean.instancesOfEAMRelationToCopy und DataObjectManagedBean.metaObject.

void copyOneInstance ()

Überprüft, ob ein Parameter "copyId" gesetzt wurde, und speichert die ID des DataObjekts, damit sie kopiert werden kann.

Definiert in Zeile 218 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.dataObject, DataObjectManagedBean.instanceCopied, DataObjectManagedBean.instanceOfEAMObjectToCopy,

DataObjectManagedBean.instanceOfEAMRelationToCopy und
DataObjectManagedBean.metaObject.

Wird benutzt von DataObjectManagedBean.getCheckDelete().

String deleteAll ()

Löscht alle Instanzen eines Objektes oder Relation

Definiert in Zeile 715 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.currentObjects,
DataObjectManagedBean.currentRelations, DataObjectManagedBean.dataObject,
DataObjectManagedBean.metaObject, DataObjectManagedBean.password,
DataObjectManagedBean.showDataObjectFromObject(),
DataObjectManagedBean.showDataObjectFromRelation() und
DataObjectManagedBean.userId.

Wird benutzt von DataObjectManagedBean.cancelDeleteAll(),
DataObjectManagedBean.deleteAllSelected(),
DataObjectManagedBean.getCurrentObjects(),
DataObjectManagedBean.getCurrentRelations() und
DataObjectManagedBean.isDeleteAll().

void deleteAllSelected ()

Setzt den boolean-Wert deleteAll auf true, wenn der Button "alle Löschen" betätigt wurde.
Das Attribut deleteAll ist an das rendered-Attribut des Warnhinweises des Löschen aller
Instanzen gebunden.

Definiert in Zeile 178 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.deleteAll().

String getCheckDelete ()

Diese Methode wird als erstes bei der Instanzanzeige eines Objektes oder Relation
aufgerufen. Sie dient dazu, zu überprüfen, ob als Parameter eine ObjektId gesetzt wurde,
die dann anschließend gelöscht wird. Da diese Methode initial ausgeführt wird, bevor die
Übersicht der Instanzen geladen wird, wird einerseits die Methoden showDataFromObject
und showDataFromRelation ausgeführt, andererseits auch die Methode copyOneInstance
ausgeführt.

Definiert in Zeile 99 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.copyOneInstance(),
DataObjectManagedBean.instanceAvailable, DataObjectManagedBean.metaObject,
DataObjectManagedBean.password,
DataObjectManagedBean.showDataObjectFromObject() und
DataObjectManagedBean.showDataObjectFromRelation().

EAMCategory getCurrentCategory ()

Definiert in Zeile 1319 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.currentCategory.

List<EAMObject> getCurrentObjects ()

Definiert in Zeile 1251 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.currentCategory,
DataObjectManagedBean.currentObjects, DataObjectManagedBean.deleteAll(),
DataObjectManagedBean.factory und EAMStaticClass.objectDAOInterface.

List<EAMRelation> getCurrentRelations ()

Definiert in Zeile 1269 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.currentCategory,
DataObjectManagedBean.currentRelations, DataObjectManagedBean.deleteAll(),
DataObjectManagedBean.factory und EAMStaticClass.relationDAOInterface.

String getEditDataObject ()

Lädt die zu editierende Instanz

Rückgabe:

Definiert in Zeile 594 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.dataObjectList, DataObjectManagedBean.edit,
DataObjectManagedBean.editObject, DataObjectManagedBean.firstMember,
DataObjectManagedBean.instanceLock, DataObjectManagedBean.metaObject,
DataObjectManagedBean.password, DataObjectManagedBean.secondMember,
RelationDataObjectBean.setId(), RelationDataObjectBean.setName() und
DataObjectManagedBean.userId.

RelationDataObjectBean getFirstMember ()

Definiert in Zeile 1354 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.firstMember.

String getFirstMemberDataObjects ()

Läd alle möglichen Instanzen des ersten RelationMemberObjektes

Definiert in Zeile 408 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.addFirstMemberToRelation(),
DataObjectManagedBean.firstMemberMultiplicity,

DataObjectManagedBean.getSecondMemberDataObjects(),
DataObjectManagedBean.metaObject und DataObjectManagedBean.password.

String getFirstMemberMultiplicity ()

Definiert in Zeile 1295 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.firstMemberMultiplicity.

EAMSuperClass getMetaObject ()

Definiert in Zeile 1344 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.metaObject.

EAMObjectDAO getObjectDAOInterface ()

Definiert in Zeile 1327 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.objectDAOInterface.

javax.faces.component.html.HtmlInputHidden getRowCounter ()

Definiert in Zeile 1311 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.rowCounter.

javax.faces.component.html.HtmlInputHidden getRowCounter2 ()

Definiert in Zeile 1335 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.rowCounter2.

RelationDataObjectBean getSecondMember ()

Definiert in Zeile 1364 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.secondMember.

String getSecondMemberDataObjects ()

Läd alle möglichen Instanzen des zweiten RelationMemberObjektes

Definiert in Zeile 467 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.addSecondMemberToRelation(),
DataObjectManagedBean.metaObject, DataObjectManagedBean.password und
DataObjectManagedBean.secondMemberMultiplicity.

Wird benutzt von DataObjectManagedBean.getFirstMemberDataObjects().

String getSecondMemberMultiplicity ()

Definiert in Zeile 1303 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.secondMemberMultiplicity.

boolean isDeleteAll ()

Definiert in Zeile 1388 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.deleteAll().

boolean isInstanceAvailable ()

Definiert in Zeile 1397 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.instanceAvailable.

boolean isInstanceCopied ()

Definiert in Zeile 1285 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.instanceOfEAMObjectToCopy, DataObjectManagedBean.instanceOfEAMRelationToCopy und DataObjectManagedBean.metaObject.

boolean isInstancesCopied ()

Definiert in Zeile 1374 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.instancesCopied.

String newDataObject ()

Definiert in Zeile 1236 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.metaObject.

String openCategory ()

Wird ausgeführt, wenn ein Metamodell ausgewählt wurde.

Definiert in Zeile 578 der Datei DataObjectManagedBean.java.

Benutzt EAMStaticClass.categoryDAOInterface, DataObjectManagedBean.currentCategory, DataObjectManagedBean.factory und DataObjectManagedBean.rowCounter.

String pasteAll ()

Fügt alle kopierten Instanzen in das jeweilige Objekt oder Relation ein.

Definiert in Zeile 335 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.dataObject, DataObjectManagedBean.factory, DataObjectManagedBean.instancesOfEAMObjectToCopy, DataObjectManagedBean.metaObject, EAMStaticClass.objectDAOInterface, DataObjectManagedBean.password, EAMStaticClass.relationDAOInterface, DataObjectManagedBean.showDataObjectFromObject() und DataObjectManagedBean.showDataObjectFromRelation().

void pasteInstance ()

Fügt eine Instanz in eine Metaobjekt ein. Sollten dies aufgrund von nicht kompatiblen Attributen fehlschlagen, wird der Benutzer darüber informiert.

Definiert in Zeile 258 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.instanceOfEAMObjectToCopy, DataObjectManagedBean.instanceOfEAMRelationToCopy, DataObjectManagedBean.metaObject, DataObjectManagedBean.password, DataObjectManagedBean.showDataObjectFromObject(), DataObjectManagedBean.showDataObjectFromRelation() und DataObjectManagedBean.userId.

String saveDataObject ()

Speichert eine Instanz eines Objektes oder Relation. Zusätzlich wird beim Speichern validiert und entsprechende Exceptions abgefangen.

Definiert in Zeile 778 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.currentObjects, DataObjectManagedBean.currentRelations, DataObjectManagedBean.dataObject, DataObjectManagedBean.dataObjectList, DataObjectManagedBean.editObject, DataObjectManagedBean.firstMember, RelationDataObjectBean.getId(), DataObjectManagedBean.instanceLock, DataObjectManagedBean.metaObject, DataObjectManagedBean.password, DataObjectManagedBean.secondMember, DataObjectManagedBean.showDataObjectFromObject(), DataObjectManagedBean.showDataObjectFromRelation() und DataObjectManagedBean.userId.

void setCurrentCategory (EAMCategory currentCategory)

Definiert in Zeile 1323 der Datei DataObjectManagedBean.java.

void setDeleteAll (boolean deleteAll)

Definiert in Zeile 1393 der Datei DataObjectManagedBean.java.

void setFirstMember (RelationDataObjectBean firstMember)

Definiert in Zeile 1359 der Datei DataObjectManagedBean.java.

void setFirstMemberMultiplicity (String firstMemberMultiplicity)

Definiert in Zeile 1299 der Datei DataObjectManagedBean.java.

void setInstanceAvailable (boolean instanceAvailable)

Definiert in Zeile 1402 der Datei DataObjectManagedBean.java.

void setInstanceCopied (boolean instanceCopied)

Definiert in Zeile 1384 der Datei DataObjectManagedBean.java.

void setInstancesCopied (boolean instancesCopied)

Definiert in Zeile 1379 der Datei DataObjectManagedBean.java.

void setMetaObject (EAMSuperClass metaObject)

Definiert in Zeile 1349 der Datei DataObjectManagedBean.java.

void setObjectDAOInterface (EAMObjectDAO objectDAOInterface)

Definiert in Zeile 1331 der Datei DataObjectManagedBean.java.

void setRowCounter (javax.faces.component.html.HtmlInputHidden rowCounter)

Definiert in Zeile 1315 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.rowCounter.

void setRowCounter2 (javax.faces.component.html.HtmlInputHidden rowCounter2)

Definiert in Zeile 1339 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.rowCounter2.

void setSecondMember (RelationDataObjectBean secondMember)

Definiert in Zeile 1369 der Datei DataObjectManagedBean.java.

void setSecondMemberMultiplicity (String secondMemberMultiplicity)

Definiert in Zeile 1307 der Datei DataObjectManagedBean.java.

String showDataObjectFromObject ()

Holt alle Instanzen eines Objektes aus der Datenbank

Definiert in Zeile 1071 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.currentObjects,
DataObjectManagedBean.dataObjectList, DataObjectManagedBean.factory,
DataObjectManagedBean.instanceAvailable, DataObjectManagedBean.metaObject,
EAMStaticClass.objectDAOInterface, DataObjectManagedBean.password und
DataObjectManagedBean.rowCounter.

Wird benutzt von DataObjectManagedBean.cancelPressed(),
DataObjectManagedBean.deleteAll(), DataObjectManagedBean.getCheckDelete(),
DataObjectManagedBean.pasteAll(), DataObjectManagedBean.pasteInstance() und
DataObjectManagedBean.saveDataObject().

String showDataObjectFromRelation ()

Holt alle Relationen eines Objektes aus der Datenbank

Definiert in Zeile 1158 der Datei DataObjectManagedBean.java.

Benutzt DataObjectManagedBean.currentRelations,
DataObjectManagedBean.dataObjectList, DataObjectManagedBean.factory,
DataObjectManagedBean.instanceAvailable, DataObjectManagedBean.metaObject,
DataObjectManagedBean.password, EAMStaticClass.relationDAOInterface und
DataObjectManagedBean.rowCounter2.

Wird benutzt von DataObjectManagedBean.cancelPressed(),
DataObjectManagedBean.deleteAll(), DataObjectManagedBean.getCheckDelete(),
DataObjectManagedBean.pasteAll(), DataObjectManagedBean.pasteInstance() und
DataObjectManagedBean.saveDataObject().

4.2.1.3 Dokumentation der Datenelemente**String checkDelete [private]**

Definiert in Zeile 77 der Datei DataObjectManagedBean.java.

EAMCategory currentCategory [private]

Definiert in Zeile 67 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.getCurrentCategory(),
DataObjectManagedBean.getCurrentObjects(),
DataObjectManagedBean.getCurrentRelations() und
DataObjectManagedBean.openCategory().

List<EAMObject> currentObjects [private]

Definiert in Zeile 68 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.deleteAll(),
DataObjectManagedBean.getCurrentObjects(),
DataObjectManagedBean.saveDataObject() und
DataObjectManagedBean.showDataObjectFromObject().

List<EAMRelation> currentRelations [private]

Definiert in Zeile 69 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.deleteAll(),
DataObjectManagedBean.getCurrentRelations(),
DataObjectManagedBean.saveDataObject() und
DataObjectManagedBean.showDataObjectFromRelation().

DataObject dataObject

Definiert in Zeile 71 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.addFirstMemberToRelation(),
DataObjectManagedBean.addSecondMemberToRelation(),
DataObjectManagedBean.copyOneInstance(), DataObjectManagedBean.deleteAll(),
DataObjectManagedBean.pasteAll() und DataObjectManagedBean.saveDataObject().

DataObjectList dataObjectList = null

Definiert in Zeile 63 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.getEditDataObject(),
DataObjectManagedBean.saveDataObject(),
DataObjectManagedBean.showDataObjectFromObject() und
DataObjectManagedBean.showDataObjectFromRelation().

boolean deleteAll = false [private]

Definiert in Zeile 86 der Datei DataObjectManagedBean.java.

boolean edit [private]

Definiert in Zeile 72 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.getEditDataObject().

String editDataObject [private]

Definiert in Zeile 78 der Datei DataObjectManagedBean.java.

DataObject editObject

Definiert in Zeile 70 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.cancelPressed(),
DataObjectManagedBean.getEditDataObject() und
DataObjectManagedBean.saveDataObject().

**EAMStaticClass factory = EAMStaticClass.getInstance(userId, password,
this.getClass().getName()) [private]**

Definiert in Zeile 61 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.getCurrentObjects(),
DataObjectManagedBean.getCurrentRelations(),
DataObjectManagedBean.openCategory(), DataObjectManagedBean.pasteAll(),
DataObjectManagedBean.showDataObjectFromObject() und
DataObjectManagedBean.showDataObjectFromRelation().

RelationDataObjectBean firstMember [private]

Definiert in Zeile 73 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.addFirstMemberToRelation(),
DataObjectManagedBean.cancelPressed(),
DataObjectManagedBean.getEditDataObject(),
DataObjectManagedBean.getFirstMember() und
DataObjectManagedBean.saveDataObject().

String firstMemberDataObjects [private]

Definiert in Zeile 79 der Datei DataObjectManagedBean.java.

String firstMemberMultiplicity [private]

Definiert in Zeile 75 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.getFirstMemberDataObjects() und
DataObjectManagedBean.getFirstMemberMultiplicity().

boolean instanceAvailable = true [private]

Definiert in Zeile 87 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.getCheckDelete(),
DataObjectManagedBean.isInstanceAvailable(),
DataObjectManagedBean.showDataObjectFromObject() und
DataObjectManagedBean.showDataObjectFromRelation().

boolean instanceCopied = false [private]

Definiert in Zeile 85 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.copyOneInstance().

Lock instanceLock = null [static, private]

Definiert in Zeile 89 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.cancelPressed(),
DataObjectManagedBean.getEditDataObject() und
DataObjectManagedBean.saveDataObject().

DataObject instanceOfEAMObjectToCopy [private]

Definiert in Zeile 82 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.copyOneInstance(),
DataObjectManagedBean.isInstanceCopied() und
DataObjectManagedBean.pasteInstance().

DataObject instanceOfEAMRelationToCopy [private]

Definiert in Zeile 83 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.copyOneInstance(),
DataObjectManagedBean.isInstanceCopied() und
DataObjectManagedBean.pasteInstance().

boolean instancesCopied = false [private]

Definiert in Zeile 84 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.copyAll() und
DataObjectManagedBean.isInstancesCopied().

String instancesOfEAMObjectToCopy [private]

Definiert in Zeile 80 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.copyAll() und
DataObjectManagedBean.pasteAll().

String instancesOfEAMRelationToCopy [private]

Definiert in Zeile 81 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.copyAll().

EAMSuperClass metaObject

Definiert in Zeile 64 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.cancelPressed(),
DataObjectManagedBean.copyAll(), DataObjectManagedBean.copyOneInstance(),
DataObjectManagedBean.deleteAll(), DataObjectManagedBean.getCheckDelete(),
DataObjectManagedBean.getEditDataObject(),
DataObjectManagedBean.getFirstMemberDataObjects(),
DataObjectManagedBean.getMetaObject(),
DataObjectManagedBean.getSecondMemberDataObjects(),
DataObjectManagedBean.isInstanceCopied(),
DataObjectManagedBean.newDataObject(), DataObjectManagedBean.pasteAll(),
DataObjectManagedBean.pasteInstance(), DataObjectManagedBean.saveDataObject(),
DataObjectManagedBean.showDataObjectFromObject() und
DataObjectManagedBean.showDataObjectFromRelation().

**EAMObjectDAO objectDAOInterface = DAOFactory.getAuthInstance(userId,
password).getEAMObjectDAO()**

Definiert in Zeile 62 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.getObjectDAOInterface().

**String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")**

Definiert in Zeile 59 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.deleteAll(),
DataObjectManagedBean.getCheckDelete(),
DataObjectManagedBean.getEditDataObject(),
DataObjectManagedBean.getFirstMemberDataObjects(),
DataObjectManagedBean.getSecondMemberDataObjects(),
DataObjectManagedBean.pasteAll(), DataObjectManagedBean.pasteInstance(),
DataObjectManagedBean.saveDataObject(),
DataObjectManagedBean.showDataObjectFromObject() und
DataObjectManagedBean.showDataObjectFromRelation().

javax.faces.component.html.HtmlInputHidden rowCounter [private]

Definiert in Zeile 65 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.getRowCounter(),
DataObjectManagedBean.openCategory(), DataObjectManagedBean.setRowCounter()
und DataObjectManagedBean.showDataObjectFromObject().

javax.faces.component.html.HtmlInputHidden rowCounter2 [private]

Definiert in Zeile 66 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.getRowCounter2(),
DataObjectManagedBean.setRowCounter2() und
DataObjectManagedBean.showDataObjectFromRelation().

RelationDataObjectBean secondMember [private]

Definiert in Zeile 74 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.addSecondMemberToRelation(),
DataObjectManagedBean.cancelPressed(),
DataObjectManagedBean.getEditDataObject(),
DataObjectManagedBean.getSecondMember() und
DataObjectManagedBean.saveDataObject().

String secondMemberMultiplicity [private]

Definiert in Zeile 76 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.getSecondMemberDataObjects() und
DataObjectManagedBean.getSecondMemberMultiplicity().

String userId =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Definiert in Zeile 58 der Datei DataObjectManagedBean.java.

Wird benutzt von DataObjectManagedBean.deleteAll(),
DataObjectManagedBean.getEditDataObject(), DataObjectManagedBean.pasteInstance()
und DataObjectManagedBean.saveDataObject().

4.2.2 QueryResultBean Klassenreferenz

4.2.2.1 Ausführliche Beschreibung

Dient zum Austausch zwischen den Tags

Autor:

Roland Koppe
David Heimann

Definiert in Zeile 15 der Datei QueryResultBean.java.

Öffentliche Methoden

- `DataObject getDataObject ()`
- `DataObjectList getDataObjectList ()`
- `String getFetchTime ()`
- `DataObjectList getFirstRelationMemberList ()`
- `EAMSuperClass getMetaObject ()`
- `DataObjectList getSecondRelationMemberList ()`
- `void setDataObject (DataObject dataObject)`
- `void setDataObjectList (DataObjectList dataObjectList)`
- `void setFetchTime (String fetchTime)`
- `void setFirstRelationMemberList (DataObjectList firstRelationMemberList)`
- `void setMetaObject (EAMSuperClass metaObject)`
- `void setSecondRelationMemberList (DataObjectList secondRelationMemberList)`

Private Attribute

- `DataObject dataObject`
- `DataObjectList dataObjectList`
- `String fetchTime`
- `DataObjectList firstRelationMemberList`
- `EAMSuperClass metaObject`
- `DataObjectList secondRelationMemberList`

4.2.2.2 Dokumentation der Elementfunktionen

DataObject getDataObject ()

Definiert in Zeile 65 der Datei `QueryResultBean.java`.

Benutzt `QueryResultBean.dataObject`.

DataObjectList getDataObjectList ()

Definiert in Zeile 41 der Datei `QueryResultBean.java`.

Benutzt `QueryResultBean.dataObjectList`.

String getFetchTime ()

Definiert in Zeile 49 der Datei `QueryResultBean.java`.

Benutzt `QueryResultBean.fetchTime`.

DataObjectList getFirstRelationMemberList ()

Definiert in Zeile 25 der Datei QueryResultBean.java.

Benutzt QueryResultBean.firstRelationMemberList.

EAMSuperClass getMetaObject ()

Definiert in Zeile 57 der Datei QueryResultBean.java.

Benutzt QueryResultBean.metaObject.

DataObjectList getSecondRelationMemberList ()

Definiert in Zeile 33 der Datei QueryResultBean.java.

Benutzt QueryResultBean.secondRelationMemberList.

void setDataObject (DataObject dataObject)

Definiert in Zeile 69 der Datei QueryResultBean.java.

void setDataObjectList (DataObjectList dataObjectList)

Definiert in Zeile 45 der Datei QueryResultBean.java.

void setFetchTime (String fetchTime)

Definiert in Zeile 53 der Datei QueryResultBean.java.

void setFirstRelationMemberList (DataObjectList firstRelationMemberList)

Definiert in Zeile 29 der Datei QueryResultBean.java.

void setMetaObject (EAMSuperClass metaObject)

Definiert in Zeile 61 der Datei QueryResultBean.java.

void setSecondRelationMemberList (DataObjectList secondRelationMemberList)

Definiert in Zeile 37 der Datei QueryResultBean.java.

4.2.2.3 Dokumentation der Datenelemente

DataObject dataObject [private]

Definiert in Zeile 19 der Datei QueryResultBean.java.

Wird benutzt von QueryResultBean.getDataObject().

DataObjectList dataObjectList [private]

Definiert in Zeile 17 der Datei QueryResultBean.java.

Wird benutzt von QueryResultBean.getDataObjectList().

String fetchTime [private]

Definiert in Zeile 22 der Datei QueryResultBean.java.

Wird benutzt von QueryResultBean.getFetchTime().

DataObjectList firstRelationMemberList [private]

Definiert in Zeile 20 der Datei QueryResultBean.java.

Wird benutzt von QueryResultBean.getFirstRelationMemberList().

EAMSuperClass metaObject [private]

Definiert in Zeile 18 der Datei QueryResultBean.java.

Wird benutzt von QueryResultBean.getMetaObject().

DataObjectList secondRelationMemberList [private]

Definiert in Zeile 21 der Datei QueryResultBean.java.

Wird benutzt von QueryResultBean.getSecondRelationMemberList().

4.2.3 RelationDataObjectBean Klassenreferenz

4.2.3.1 Ausführliche Beschreibung

Bean zur Darstellung von Relationsmitgliedern

Autor:

David Heimann

Definiert in Zeile 9 der Datei RelationDataObjectBean.java.

Öffentliche Methoden

- long getId ()
- Object getName ()
- void setId (long id)
- void setName (Object name)

Private Attribute

- long id
- Object name

4.2.3.2 Dokumentation der Elementfunktionen

long getId ()

Definiert in Zeile 20 der Datei RelationDataObjectBean.java.

Benutzt RelationDataObjectBean.id.

Wird benutzt von DataObjectManagedBean.saveDataObject().

Object getName ()

Definiert in Zeile 14 der Datei RelationDataObjectBean.java.

Benutzt RelationDataObjectBean.name.

void setId (long id)

Definiert in Zeile 23 der Datei RelationDataObjectBean.java.

Wird benutzt von DataObjectManagedBean.addFirstMemberToRelation(), DataObjectManagedBean.addSecondMemberToRelation() und DataObjectManagedBean.getEditDataObject().

void setName (Object name)

Definiert in Zeile 17 der Datei RelationDataObjectBean.java.

Wird benutzt von DataObjectManagedBean.addFirstMemberToRelation(), DataObjectManagedBean.addSecondMemberToRelation() und DataObjectManagedBean.getEditDataObject().

4.2.3.3 Dokumentation der Datenelemente

long id [private]

Definiert in Zeile 12 der Datei RelationDataObjectBean.java.

Wird benutzt von RelationDataObjectBean.getId().

Object name [private]

Definiert in Zeile 11 der Datei RelationDataObjectBean.java.

Wird benutzt von RelationDataObjectBean.getName().

4.3 Tag-Library

4.3.1 DataObjectScrollerTag Klassenreferenz

Abgeleitet von javax::servlet::jsp::tagext::TagSupport.

4.3.1.1 Ausführliche Beschreibung

Das DataObjectScrollerTag stellt einen Scroller für dargestellte DataObjects durch das DataObjectTableTag dar.

Autor:

Roland Koppe

Definiert in Zeile 21 der Datei DataObjectScrollerTag.java.

Öffentliche Methoden

- int doAfterBody () throws JspException
- int doEndTag () throws JspException
- int doStartTag () throws JspException
- DataObjectList getDataObjectList ()
- String getRedirect ()
- String getScrollerId ()
- void setDataObjectList (DataObjectList dataObjectList)
- void setIconFirst (String iconFirst)
- void setIconLast (String iconLast)
- void setIconNext (String iconNext)
- void setIconPrevious (String iconPrevious)
- void setPageContext (PageContext pageContext)
- void setRedirect (String redirectURI)
- void setScrollerId (String id)

Statische öffentliche Attribute

- static final String ID_ACTION = "action"
- static final String ID_FIRST = "first"
- static final String ID_LAST = "last"
- static final String ID_NEXT = "next"
- static final String ID_PAGE = "page"
- static final String ID_PREVIOUS = "previous"

Geschützte Attribute

- String redirectURI

Private Methoden

- String createLink (String id, int page)

Private Attribute

- DataObjectList dataObjectList
- String iconFirst
- String iconLast
- String iconNext
- String iconPrevious
- String scrollerId

Statische private Attribute

- static final long serialVersionUID = 1L

4.3.1.2 Dokumentation der Elementfunktionen

String createLink (String id, int page) [private]

Definiert in Zeile 281 der Datei DataObjectScrollerTag.java.

Benutzt DataObjectScrollerTag.ID_ACTION, DataObjectScrollerTag.ID_PAGE, DataObjectScrollerTag.redirectURI und DataObjectScrollerTag.scrollerId.

Wird benutzt von DataObjectScrollerTag.doStartTag().

int doAfterBody () throws JspException

Definiert in Zeile 156 der Datei DataObjectScrollerTag.java.

int doEndTag () throws JspException

Definiert in Zeile 163 der Datei DataObjectScrollerTag.java.

int doStartTag () throws JspException

Schreibt den Scroller.

Definiert in Zeile 173 der Datei DataObjectScrollerTag.java.

Benutzt `DataObjectScrollerTag.createLink()`, `DataObjectScrollerTag.dataObjectList`, `DataObjectScrollerTag.iconFirst`, `DataObjectScrollerTag.iconLast`, `DataObjectScrollerTag.iconNext`, `DataObjectScrollerTag.iconPrevious`, `DataObjectScrollerTag.ID_FIRST`, `DataObjectScrollerTag.ID_LAST`, `DataObjectScrollerTag.ID_NEXT`, `DataObjectScrollerTag.ID_PAGE`, `DataObjectScrollerTag.ID_PREVIOUS` und `DataObjectScrollerTag.scrollerId`.

DataObjectList getDataObjectList ()

Liefert die `DataObjectList` des Scrollers.

Rückgabe:

`DataObjectList`

Definiert in Zeile 72 der Datei `DataObjectScrollerTag.java`.

Benutzt `DataObjectScrollerTag.dataObjectList`.

String getRedirect ()

Liefert die URL des Redirects, nach dem Betätigen eines Knopfs.

Rückgabe:

`String` URI

Definiert in Zeile 138 der Datei `DataObjectScrollerTag.java`.

Benutzt `DataObjectScrollerTag.redirectURL`.

String getScrollerId ()

Liefert die HTML-ID des Scrollers.

Rückgabe:

`String` ID des Scrollers

Definiert in Zeile 50 der Datei `DataObjectScrollerTag.java`.

Benutzt `DataObjectScrollerTag.scrollerId`.

void setDataObjectList (DataObjectList dataObjectList)

Setzt die `DataObjectList` für den Scroller.

Parameter:

dataObjectList `DataObjectList`

Definiert in Zeile 83 der Datei `DataObjectScrollerTag.java`.

void setIconFirst (String iconFirst)

Setzt die URL für das Bild von "ID_FIRST".

Parameter:

iconFirst String URL

Definiert in Zeile 94 der Datei DataObjectScrollerTag.java.

void setIconLast (String iconLast)

Setzt die URL für das Bild von "ID_LAST".

Parameter:

iconPrevious String URL

Definiert in Zeile 127 der Datei DataObjectScrollerTag.java.

void setIconNext (String iconNext)

Setzt die URL für das Bild von "ID_NEXT".

Parameter:

iconNext String URL

Definiert in Zeile 116 der Datei DataObjectScrollerTag.java.

void setIconPrevious (String iconPrevious)

Setzt die URL für das Bild von "ID_PREVIOUS".

Parameter:

iconPrevious String URL

Definiert in Zeile 105 der Datei DataObjectScrollerTag.java.

void setPageContext (PageContext pageContext)

Gibt den JavaScript-Ausdruck für das onclick-Ereignis der Knöpfe zurück.

Parameter:

name String ID Namen, Konstanten

Rückgabe:

String onclick JavaScript Gibt den JavaScript-Ausdruck für das onclick-Ereignis der Seitenzahl zurück.

Parameter:

page int Seitenzahl

Rückgabe:

String onclick JavaScript

Definiert in Zeile 325 der Datei DataObjectScrollerTag.java.

void setRedirect (String redirectURI)

Setzt die URL für den Redirect.

Parameter:

redirectURI String URI

Definiert in Zeile 149 der Datei DataObjectScrollerTag.java.

void setScrollerId (String id)

Setzt die HTML-ID des Scrollers.

Parameter:

id String ID des Scrollers

Definiert in Zeile 61 der Datei DataObjectScrollerTag.java.

Benutzt DataObjectScrollerTag.scrollerId.

4.3.1.3 Dokumentation der Datenelemente**DataObjectList dataObjectList [private]**

Definiert in Zeile 34 der Datei DataObjectScrollerTag.java.

Wird benutzt von DataObjectScrollerTag.doStartTag() und DataObjectScrollerTag.getDataObjectList().

String iconFirst [private]

Definiert in Zeile 38 der Datei DataObjectScrollerTag.java.

Wird benutzt von DataObjectScrollerTag.doStartTag().

String iconLast [private]

Definiert in Zeile 41 der Datei DataObjectScrollerTag.java.

Wird benutzt von DataObjectScrollerTag.doStartTag().

String iconNext [private]

Definiert in Zeile 40 der Datei DataObjectScrollerTag.java.

Wird benutzt von DataObjectScrollerTag.doStartTag().

String iconPrevious [private]

Definiert in Zeile 39 der Datei DataObjectScrollerTag.java.

Wird benutzt von DataObjectScrollerTag.doStartTag().

final String ID_ACTION = "action" [static]

Definiert in Zeile 25 der Datei DataObjectScrollerTag.java.

Wird benutzt von DataObjectScrollerTag.createLink().

final String ID_FIRST = "first" [static]

Definiert in Zeile 26 der Datei DataObjectScrollerTag.java.

Wird benutzt von DataObjectScrollerTag.doStartTag().

final String ID_LAST = "last" [static]

Definiert in Zeile 29 der Datei DataObjectScrollerTag.java.

Wird benutzt von DataObjectScrollerTag.doStartTag().

final String ID_NEXT = "next" [static]

Definiert in Zeile 28 der Datei DataObjectScrollerTag.java.

Wird benutzt von DataObjectScrollerTag.doStartTag().

final String ID_PAGE = "page" [static]

Definiert in Zeile 30 der Datei DataObjectScrollerTag.java.

Wird benutzt von DataObjectScrollerTag.createLink() und DataObjectScrollerTag.doStartTag().

final String ID_PREVIOUS = "previous" [static]

Definiert in Zeile 27 der Datei DataObjectScrollerTag.java.

Wird benutzt von DataObjectScrollerTag.doStartTag().

String redirectURI [protected]

Definiert in Zeile 32 der Datei DataObjectScrollerTag.java.

Wird benutzt von DataObjectScrollerTag.createLink() und DataObjectScrollerTag.getRedirect().

String scrollerId [private]

Definiert in Zeile 36 der Datei DataObjectScrollerTag.java.

Wird benutzt von DataObjectScrollerTag.createLink(),
DataObjectScrollerTag.doStartTag(), DataObjectScrollerTag.getScrollerId() und
DataObjectScrollerTag.setScrollerId().

final long serialVersionUID = 1L [static, private]

Definiert in Zeile 23 der Datei DataObjectScrollerTag.java.

4.3.2 dataObjectTable Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::instance::taglib::DataObjectTableTag.

4.3.2.1 Ausführliche Beschreibung

Tag zur Darstellung von Instanzen

Autor:

David Heimann

Definiert in Zeile 35 der Datei dataObjectTable.java.

Öffentliche Methoden

- int doAfterBody () throws JspException
- int doEndTag () throws JspException
- int doStartTag () throws JspException
- String getType ()
- void setPageContext (PageContext pageContext)
- void setType (String type)

Private Methoden

- boolean implementationOfRoletag (String requireRole)
- ArrayList< String > parseRole (String requireRole)

Private Attribute

- String type

Statische private Attribute

- static final long serialVersionUID = 1L

4.3.2.2 Dokumentation der Elementfunktionen

int doAfterBody () throws JspException

Definiert in Zeile 56 der Datei dataObjectTable.java.

int doEndTag () throws JspException

Definiert in Zeile 62 der Datei dataObjectTable.java.

int doStartTag () throws JspException

Gibt eine Tabelle mit Instanzen aus.

Autor:

David Heimann

Definiert in Zeile 73 der Datei dataObjectTable.java.

Benutzt dataObjectTable.implementationOfRoletag() und dataObjectTable.type.

String getType ()

Definiert in Zeile 40 der Datei dataObjectTable.java.

Benutzt dataObjectTable.type.

boolean implementationOfRoletag (String requireRole) [private]

Definiert in Zeile 309 der Datei dataObjectTable.java.

Benutzt dataObjectTable.parseRole().

Wird benutzt von dataObjectTable.doStartTag().

ArrayList<String> parseRole (String requireRole) [private]

Definiert in Zeile 346 der Datei dataObjectTable.java.

Wird benutzt von dataObjectTable.implementationOfRoletag().

void setPageContext (PageContext pageContext)

Definiert in Zeile 305 der Datei dataObjectTable.java.

void setType (String type)

Definiert in Zeile 45 der Datei dataObjectTable.java.

4.3.2.3 Dokumentation der Datenelemente**final long serialVersionUID = 1L** [static, private]

Definiert in Zeile 53 der Datei dataObjectTable.java.

String type [private]

Definiert in Zeile 37 der Datei dataObjectTable.java.

Wird benutzt von dataObjectTable.doStartTag() und dataObjectTable.getType().

4.3.3 insertDataObject Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::metamodel::instance::taglib::DataObjectTableTag.

4.3.3.1 Ausführliche Beschreibung

Gibt eine Tabelle mit den Attributen einer Instanz wieder

Autor:

David Heimann

Definiert in Zeile 29 der Datei insertDataObject.java.

Öffentliche Methoden

- int doAfterBody () throws JspException
- int doEndTag () throws JspException
- int doStartTag () throws JspException
- DataObject getDataObject ()
- EAMSuperClass getMetaObject ()
- void setDataObject (DataObject dataObject)
- void setMetaObject (EAMSuperClass metaObject)
- void setPageContext (PageContext pageContext)

Private Attribute

- DataObject dataObject
- EAMSuperClass metaObject

Statische private Attribute

- static final long serialVersionUID = 1L

4.3.3.2 Dokumentation der Elementfunktionen

int doAfterBody () throws JspException

Definiert in Zeile 45 der Datei insertDataObject.java.

int doEndTag () throws JspException

Definiert in Zeile 50 der Datei insertDataObject.java.

int doStartTag () throws JspException

Definiert in Zeile 55 der Datei insertDataObject.java.

Benutzt insertDataObject.dataObject und insertDataObject.metaObject.

DataObject getDataObject ()

Definiert in Zeile 154 der Datei insertDataObject.java.

Benutzt insertDataObject.dataObject.

EAMSuperClass getMetaObject ()

Definiert in Zeile 36 der Datei insertDataObject.java.

Benutzt insertDataObject.metaObject.

void setDataObject (DataObject dataObject)

Definiert in Zeile 158 der Datei insertDataObject.java.

void setMetaObject (EAMSuperClass metaObject)

Definiert in Zeile 40 der Datei insertDataObject.java.

void setPageContext (PageContext pageContext)

Definiert in Zeile 163 der Datei insertDataObject.java.

4.3.3.3 Dokumentation der Datenelemente

DataObject dataObject [private]

Definiert in Zeile 34 der Datei insertDataObject.java.

Wird benutzt von insertDataObject.doStartTag() und insertDataObject.getDataObject().

EAMSuperClass metaObject [private]

Definiert in Zeile 33 der Datei insertDataObject.java.

Wird benutzt von insertDataObject.doStartTag() und insertDataObject.getMetaObject().

final long serialVersionUID = 1L [static, private]

Definiert in Zeile 31 der Datei insertDataObject.java.

4.3.4 MessageTag Klassenreferenz

Abgeleitet von javax::servlet::jsp::tagext::TagSupport.

4.3.4.1 Ausführliche Beschreibung

Diese Klasse ist fuer die Taglibrary des Sicherheitssystem. Mit Hilfe dieses Tags werden Bereiche auf einer Seite mit einer Rolle/ID/Recht gekennzeichnet. Diese/s Rolle/ID/Recht wird spaeter von Admin dann auf Rollen gemappt die existieren.

Autor:

christian zillmann

Definiert in Zeile 22 der Datei MessageTag.java.

Öffentliche Methoden

- int doAfterBody () throws JspException
- int doEndTag () throws JspException
- int doStartTag () throws JspException
- void setMessageAttribut (String messageAttribut)
- void setPageContext (PageContext pageContext)

Private Attribute

- String messageAttribut

Statische private Attribute

- static final long serialVersionUID = 1L

4.3.4.2 Dokumentation der Elementfunktionen

int doAfterBody () throws JspException

Definiert in Zeile 43 der Datei MessageTag.java.

int doEndTag () throws JspException

Definiert in Zeile 54 der Datei MessageTag.java.

int doStartTag () throws JspException

Definiert in Zeile 63 der Datei MessageTag.java.

Benutzt MessageTag.messageAttribut.

void setMessageAttribut (String messageAttribut)

Definiert in Zeile 33 der Datei MessageTag.java.

void setPageContext (PageContext pageContext)

Definiert in Zeile 92 der Datei MessageTag.java.

4.3.4.3 Dokumentation der Datenelemente

String messageAttribut [private]

Definiert in Zeile 29 der Datei MessageTag.java.

Wird benutzt von MessageTag.doStartTag().

final long serialVersionUID = 1L [static, private]

Definiert in Zeile 27 der Datei MessageTag.java.

4.4 Metamodelleingabe

4.4.1 EAMAttributeManagedBean Klassenreferenz

4.4.1.1 Ausführliche Beschreibung

ManagedBean für alle Aktionen mit Attribute bzw. Attributtypen

Autor:

David Heimann
Hyung-Bin Kim

Definiert in Zeile 24 der Datei EAMAttributeManagedBean.java.

Öffentliche Methoden

- String addAttributeDataType ()
- String addAttributeType ()
- String addExistingAttribute ()
- String backToAttribute ()
- String backToAttributeType ()
- String cancelTempAttribute ()
- String checkDeleteAttribute ()
- String checkDeleteType ()
- String deleteAttribute () throws Exception
- String deleteAttributeType () throws Exception
- String editAttribute ()
- String editAttributeType ()
- String existingAttribute ()
- int getAttrCount ()
- javax.faces.component.html.HtmlInputHidden getAttributeCounter ()
- javax.faces.component.html.HtmlInputHidden getAttributeCounter2 ()
- javax.faces.component.html.HtmlInputHidden getAttributeDataTypeCounter ()
- javax.faces.component.html.HtmlInputHidden getAttributeTypeCounter ()
- long getCheckDelete ()
- ArrayList< EAMAttribute > getCurrentAttributes ()
- EAMAttributeType getCurrentAttributeName ()
- ArrayList< EAMAttribute > getExistingAttributes ()
- ArrayList< DataType > getPossibleAttributeDataTypes ()
- ArrayList< EAMAttributeType > getPossibleAttributeTypes ()
- String getSortColumnOver ()
- EAMAttribute getTempAttribute ()
- EAMAttributeType getTempAttributeType ()
- boolean isAttributeTypeSelected ()
- boolean isCheckDeleteSelected ()
- boolean isCreateExAttribute ()
- boolean isCreateNewAttribute ()
- boolean isCreateNewAttributeType ()
- boolean isExAttributeSelected ()
- boolean isSortAscendingOver ()

- boolean isTempAttributeSelected ()
- boolean isTempAttributeTypeSelected ()
- String newAttribute ()
- String newAttributeType ()
- String noDelete ()
- String saveAttribute () throws Exception
- String saveAttributeType () throws Exception
- void setAttrCount (int attrCount)
- void setAttributeCounter (javax.faces.component.html.HtmlInputHidden attributeCounter)
- void setAttributeCounter2 (javax.faces.component.html.HtmlInputHidden attributeCounter2)
- void setAttributeDataTypeCounter (javax.faces.component.html.HtmlInputHidden attributeDataTypeCounter)
- void setAttributeTypeCounter (javax.faces.component.html.HtmlInputHidden attributeTypeCounter)
- void setAttributeTypeSelected (boolean attributeTypeSelected)
- void setCheckDelete (long checkDelete)
- void setCheckDeleteSelected (boolean checkDeleteSelected)
- void setCreateExAttribute (boolean createExAttribute)
- void setCreateNewAttribute (boolean createNewAttribute)
- void setCreateNewAttributeType (boolean createNewAttributeType)
- void setCurrentAttributes (ArrayList< EAMAttribute > currentAttributes)
- void setCurrentAttributeTypeName (EAMAttributeType currentAttributeTypeName)
- void setExAttributeSelected (boolean exAttributeSelected)
- void setExistingAttributes (ArrayList< EAMAttribute > existingAttributes)
- void setPossibleAttributeDataTypes (ArrayList< DataType > possibleAttributeDataTypes)
- void setPossibleAttributeTypes (ArrayList< EAMAttributeType > possibleAttributeTypes)
- void setSortAscendingOver (boolean sortAscendingOver)
- void setSortColumnOver (String sortColumnOver)
- void setTempAttribute (EAMAttribute tempAttribute)
- void setTempAttributeSelected (boolean tempAttributeSelected)
- void setTempAttributeType (EAMAttributeType tempAttributeType)
- void setTempAttributeTypeSelected (boolean tempAttributeTypeSelected)

Öffentliche Attribute

- `ArrayList< EAMAttribute > currentAttributes = new ArrayList<EAMAttribute>()`
- `ArrayList< EAMAttribute > existingAttributes = new ArrayList<EAMAttribute>()`
- `String password = (String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")`
- `String userId = (String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")`

Private Attribute

- `int attrCount = 0`
- `javax.faces.component.html.HtmlInputHidden attributeCounter`
- `javax.faces.component.html.HtmlInputHidden attributeCounter2`
- `javax.faces.component.html.HtmlInputHidden attributeDataTypeCounter`
- `javax.faces.component.html.HtmlInputHidden attributeTypeCounter`
- `boolean attributeTypeSelected`
- `long checkDelete`
- `boolean checkDeleteSelected`
- `boolean createExAttribute = true`
- `boolean createNewAttribute = true`
- `boolean createNewAttributeType = true`
- `EAMAttributeType currentAttributeName`
- `boolean exAttributeSelected`
- `EAMStaticClass factory = EAMStaticClass.getInstance(userId, password, this.getClass().getName())`
- `ArrayList< DataType > possibleAttributeDataTypes = new ArrayList<DataType>()`
- `ArrayList< EAMAttributeType > possibleAttributeTypes = new ArrayList<EAMAttributeType>()`
- `int saveCounter = 0`
- `boolean sortAscendingOver = true`
- `String sortColumnOver = null`
- `EAMAttribute tempAttribute`
- `boolean tempAttributeSelected`
- `EAMAttributeType tempAttributeType`
- `boolean tempAttributeTypeSelected`

4.4.1.2 Dokumentation der Elementfunktionen

String addAttributeDataType ()

Definiert in Zeile 608 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeDataTypeCounter, EAMAttributeManagedBean.possibleAttributeDataTypes und EAMAttributeManagedBean.tempAttributeType.

String addAttributeType ()

Definiert in Zeile 457 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeTypeCounter, EAMAttributeManagedBean.possibleAttributeTypes und EAMAttributeManagedBean.tempAttribute.

String addExistingAttribute ()

Fügt ein bereits existierendes Attribut einem Objekt/Relation an.

Rückgabe:

Definiert in Zeile 288 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeCounter2, EAMAttributeManagedBean.existingAttributes und EAMAttributeManagedBean.tempAttribute.

String backToAttribute ()

Definiert in Zeile 646 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeTypeSelected, EAMAttributeManagedBean.createExAttribute, EAMAttributeManagedBean.createNewAttribute und EAMAttributeManagedBean.tempAttributeSelected.

String backToAttributeType ()

Definiert in Zeile 659 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.createNewAttributeType und EAMAttributeManagedBean.tempAttributeTypeSelected.

Wird benutzt von EAMAttributeManagedBean.getCurrentAttributes().

String cancelTempAttribute ()

Setzt die booleans bei der Attribut JSF Seite zurück.

Rückgabe:

Definiert in Zeile 312 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeTypeSelected,
EAMAttributeManagedBean.createExAttribute,
EAMAttributeManagedBean.createNewAttribute,
EAMAttributeManagedBean.exAttributeSelected und
EAMAttributeManagedBean.tempAttributeSelected.

Wird benutzt von EAMAttributeManagedBean.getCurrentAttributes().

String checkDeleteAttribute ()

Sicherheitsabfrage beim Löschen eines Attributs.

Rückgabe:

Definiert in Zeile 328 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attrCount,
EAMAttributeManagedBean.attributeCounter,
EAMAttributeManagedBean.attributeTypeSelected,
EAMAttributeManagedBean.checkDelete,
EAMAttributeManagedBean.checkDeleteSelected,
EAMAttributeManagedBean.createExAttribute,
EAMAttributeManagedBean.createNewAttribute, EAMStaticClass.currentObjectRelation,
EAMAttributeManagedBean.exAttributeSelected, EAMAttributeManagedBean.factory,
EAMAttributeManagedBean.tempAttribute und
EAMAttributeManagedBean.tempAttributeSelected.

String checkDeleteType ()

Definiert in Zeile 580 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attrCount,
EAMAttributeManagedBean.attributeTypeCounter,
EAMStaticClass.attributeTypeDAOInterface,
EAMAttributeManagedBean.attributeTypeSelected,
EAMAttributeManagedBean.checkDelete,
EAMAttributeManagedBean.checkDeleteSelected,
EAMAttributeManagedBean.createNewAttributeType,
EAMAttributeManagedBean.factory, EAMAttributeManagedBean.tempAttributeType
und EAMAttributeManagedBean.tempAttributeTypeSelected.

String deleteAttribute () throws Exception

Löscht das Attribut aus der DAO.

Rückgabe:

Ausnahmebehandlung:

Exception

Definiert in Zeile 358 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.checkDelete,
EAMAttributeManagedBean.checkDeleteSelected,
EAMAttributeManagedBean.createExAttribute,
EAMAttributeManagedBean.createNewAttribute, EAMStaticClass.currentObjectRelation
und EAMAttributeManagedBean.factory.

String deleteAttributeType () throws Exception

Definiert in Zeile 596 der Datei EAMAttributeManagedBean.java.

Benutzt EAMStaticClass.attributeTypeDAOInterface,
EAMAttributeManagedBean.checkDelete,
EAMAttributeManagedBean.checkDeleteSelected,
EAMAttributeManagedBean.createNewAttributeType und
EAMAttributeManagedBean.factory.

String editAttribute ()

Holt die Informationen des zu editierenden Attribut aus der Factory und stellt diese dar.

Rückgabe:

Definiert in Zeile 219 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attrCount,
EAMAttributeManagedBean.attributeCounter,
EAMAttributeManagedBean.attributeTypeSelected,
EAMAttributeManagedBean.checkDeleteSelected,
EAMAttributeManagedBean.createExAttribute,
EAMAttributeManagedBean.createNewAttribute, EAMStaticClass.currentObjectRelation,
EAMAttributeManagedBean.exAttributeSelected, EAMAttributeManagedBean.factory,
EAMAttributeManagedBean.tempAttribute und
EAMAttributeManagedBean.tempAttributeSelected.

String editAttributeType ()

Holt die Informationen des zu editierenden Typs aus der DAO und stellt diese dar.

Rückgabe:

Definiert in Zeile 428 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeTypeCounter,
EAMStaticClass.attributeTypeDAOInterface,
EAMAttributeManagedBean.checkDeleteSelected,
EAMAttributeManagedBean.createNewAttributeType,
EAMAttributeManagedBean.factory, EAMAttributeManagedBean.tempAttributeType
und EAMAttributeManagedBean.tempAttributeTypeSelected.

String existingAttribute ()

Definiert in Zeile 267 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attrCount,
EAMAttributeManagedBean.attributeTypeSelected,
EAMAttributeManagedBean.checkDeleteSelected,
EAMAttributeManagedBean.createExAttribute,
EAMAttributeManagedBean.createNewAttribute,
EAMAttributeManagedBean.exAttributeSelected,
EAMAttributeManagedBean.getExistingAttributes(),
EAMAttributeManagedBean.tempAttribute und
EAMAttributeManagedBean.tempAttributeSelected.

int getAttrCount ()

Definiert in Zeile 820 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attrCount.

javax.faces.component.html.HtmlInputHidden getAttributeCounter ()

Definiert in Zeile 699 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeCounter.

javax.faces.component.html.HtmlInputHidden getAttributeCounter2 ()

Definiert in Zeile 786 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeCounter2.

javax.faces.component.html.HtmlInputHidden getAttributeDataTypeCounter ()

Definiert in Zeile 733 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeDataTypeCounter.

javax.faces.component.html.HtmlInputHidden getAttributeTypeCounter ()

Definiert in Zeile 708 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeTypeCounter.

long getCheckDelete ()

Definiert in Zeile 682 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.checkDelete.

ArrayList<EAMAttribute> getCurrentAttributes ()

Liefert alle Attribute des aktuellen Objektes/Relation.

Rückgabe:

ArrayList mit den Attributen.

Definiert in Zeile 67 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attrCount,
EAMAttributeManagedBean.backToAttributeType(),
EAMAttributeManagedBean.cancelTempAttribute(),
EAMAttributeManagedBean.currentAttributes, EAMStaticClass.currentObjectRelation,
EAMAttributeManagedBean.factory, EAMStaticClass.newRequest und
EAMAttributeManagedBean.noDelete().

EAMAttributeType getCurrentAttributeName ()

Definiert in Zeile 673 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.currentAttributeName.

ArrayList<EAMAttribute> getExistingAttributes ()

Liefert bereits vorhandene Attributen aus allen anderen Metamodellen.

Rückgabe:

ArrayList mit allen vorhandenen Attributen.

Definiert in Zeile 98 der Datei EAMAttributeManagedBean.java.

Benutzt EAMStaticClass.attributeDAOInterface,
EAMAttributeManagedBean.existingAttributes und EAMAttributeManagedBean.factory.
Wird benutzt von EAMAttributeManagedBean.existingAttribute().

ArrayList<DataType> getPossibleAttributeDataTypes ()

Listet die möglichen Datentypen für ein Attributstyp auf.

Rückgabe:

Definiert in Zeile 629 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.possibleAttributeDataTypes.

ArrayList<EAMAttributeType> getPossibleAttributeTypes ()

Zeigt die möglichen Attributtypen die vorhanden sind aus.

Definiert in Zeile 403 der Datei EAMAttributeManagedBean.java.

Benutzt EAMStaticClass.attributeTypeDAOInterface,

EAMAttributeManagedBean.factory und

EAMAttributeManagedBean.possibleAttributeTypes.

String getSortColumnOver ()

Definiert in Zeile 828 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.sortColumnOver.

EAMAttribute getTempAttribute ()

Definiert in Zeile 717 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.tempAttribute.

EAMAttributeType getTempAttributeType ()

Definiert in Zeile 725 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.tempAttributeType.

boolean isAttributeTypeSelected ()

Definiert in Zeile 778 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeTypeSelected.

boolean isCheckDeleteSelected ()

Definiert in Zeile 750 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.checkDeleteSelected.

boolean isCreateExAttribute ()

Definiert in Zeile 796 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.createExAttribute.

boolean isCreateNewAttribute ()

Definiert in Zeile 804 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.createNewAttribute.

boolean isCreateNewAttributeType ()

Definiert in Zeile 812 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.createNewAttributeType.

boolean isExAttributeSelected ()

Definiert in Zeile 770 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.exAttributeSelected.

boolean isSortAscendingOver ()

Definiert in Zeile 836 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.sortAscendingOver.

boolean isTempAttributeSelected ()

Definiert in Zeile 742 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.tempAttributeSelected.

boolean isTempAttributeTypeSelected ()

Definiert in Zeile 758 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.tempAttributeTypeSelected.

String newAttribute ()

Definiert in Zeile 251 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attrCount,
EAMAttributeManagedBean.attributeTypeSelected,
EAMAttributeManagedBean.checkDeleteSelected,
EAMAttributeManagedBean.createExAttribute,
EAMAttributeManagedBean.createNewAttribute,

EAMAttributeManagedBean.exAttributeSelected,
EAMAttributeManagedBean.tempAttribute und
EAMAttributeManagedBean.tempAttributeSelected.

String newAttributeType ()

Definiert in Zeile 473 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attrCount,
EAMAttributeManagedBean.checkDeleteSelected,
EAMAttributeManagedBean.createNewAttributeType,
EAMAttributeManagedBean.tempAttributeType und
EAMAttributeManagedBean.tempAttributeTypeSelected.

String noDelete ()

Definiert in Zeile 385 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.checkDeleteSelected,
EAMAttributeManagedBean.createExAttribute,
EAMAttributeManagedBean.createNewAttribute und
EAMAttributeManagedBean.createNewAttributeType.

Wird benutzt von EAMAttributeManagedBean.getCurrentAttributes().

String saveAttribute () throws Exception

Überprüft die Korrektheit der Eingaben und speichert das Attribut (auch editierte, ID>0) in die factory.

Rückgabe:

Ausnahmebehandlung:

Exception

Definiert in Zeile 144 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeTypeSelected,
EAMAttributeManagedBean.createExAttribute,
EAMAttributeManagedBean.createNewAttribute, EAMStaticClass.currentObjectRelation,
EAMAttributeManagedBean.exAttributeSelected, EAMAttributeManagedBean.factory,
EAMAttributeManagedBean.saveCounter, EAMAttributeManagedBean.tempAttribute
und EAMAttributeManagedBean.tempAttributeSelected.

String saveAttributeType () throws Exception

Überprüft die Korrektheit der Eingaben und speichert den Attributtyp (auch editierte, ID > 0) in die DAO.

Rückgabe:

Ausnahmebehandlung:

Exception

Definiert in Zeile 489 der Datei EAMAttributeManagedBean.java.

Benutzt EAMStaticClass.attributeTypeDAOInterface,
EAMAttributeManagedBean.createNewAttributeType,
EAMAttributeManagedBean.factory, EAMAttributeManagedBean.tempAttributeType
und EAMAttributeManagedBean.tempAttributeTypeSelected.

void setAttrCount (int attrCount)

Definiert in Zeile 824 der Datei EAMAttributeManagedBean.java.

**void setAttributeCounter (javax.faces.component.html.HtmlInputHidden
attributeCounter)**

Definiert in Zeile 703 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeCounter.

**void setAttributeCounter2 (javax.faces.component.html.HtmlInputHidden
attributeCounter2)**

Definiert in Zeile 790 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeCounter2.

**void setAttributeDataCounter (javax.faces.component.html.HtmlInputHidden
attributeDataCounter)**

Definiert in Zeile 737 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeDataCounter.

**void setAttributeTypeCounter (javax.faces.component.html.HtmlInputHidden
attributeTypeCounter)**

Definiert in Zeile 712 der Datei EAMAttributeManagedBean.java.

Benutzt EAMAttributeManagedBean.attributeTypeCounter.

void setAttributeTypeSelected (boolean attributeTypeSelected)

Definiert in Zeile 782 der Datei EAMAttributeManagedBean.java.

void setCheckDelete (long checkDelete)

Definiert in Zeile 686 der Datei EAMAttributeManagedBean.java.

void setCheckDeleteSelected (boolean checkDeleteSelected)

Definiert in Zeile 754 der Datei EAMAttributeManagedBean.java.

void setCreateExAttribute (boolean createExAttribute)

Definiert in Zeile 800 der Datei EAMAttributeManagedBean.java.

void setCreateNewAttribute (boolean createNewAttribute)

Definiert in Zeile 808 der Datei EAMAttributeManagedBean.java.

void setCreateNewAttributeType (boolean createNewAttributeType)

Definiert in Zeile 816 der Datei EAMAttributeManagedBean.java.

void setCurrentAttributes (ArrayList< EAMAttribute > currentAttributes)

Definiert in Zeile 694 der Datei EAMAttributeManagedBean.java.

void setCurrentAttributeName (EAMAttributeType currentAttributeName)

Definiert in Zeile 677 der Datei EAMAttributeManagedBean.java.

void setExAttributeSelected (boolean exAttributeSelected)

Definiert in Zeile 774 der Datei EAMAttributeManagedBean.java.

void setExistingAttributes (ArrayList< EAMAttribute > existingAttributes)

Definiert in Zeile 766 der Datei EAMAttributeManagedBean.java.

void setPossibleAttributeDataTypes (ArrayList< DataType > possibleAttributeDataTypes)

Definiert in Zeile 690 der Datei EAMAttributeManagedBean.java.

void setPossibleAttributeTypes (ArrayList< EAMAttributeType > possibleAttributeTypes)

Definiert in Zeile 668 der Datei EAMAttributeManagedBean.java.

void setSortAscendingOver (boolean sortAscendingOver)

Definiert in Zeile 840 der Datei EAMAttributeManagedBean.java.

void setSortColumnOver (String sortColumnOver)

Definiert in Zeile 832 der Datei EAMAttributeManagedBean.java.

void setTempAttribute (EAMAttribute tempAttribute)

Definiert in Zeile 721 der Datei EAMAttributeManagedBean.java.

void setTempAttributeSelected (boolean tempAttributeSelected)

Definiert in Zeile 746 der Datei EAMAttributeManagedBean.java.

void setTempAttributeType (EAMAttributeType tempAttributeType)

Definiert in Zeile 729 der Datei EAMAttributeManagedBean.java.

void setTempAttributeTypeSelected (boolean tempAttributeTypeSelected)

Definiert in Zeile 762 der Datei EAMAttributeManagedBean.java.

4.4.1.3 Dokumentation der Datenelemente

int attrCount = 0 [private]

Definiert in Zeile 55 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.checkDeleteAttribute(),
EAMAttributeManagedBean.checkDeleteType(),
EAMAttributeManagedBean.editAttribute(),
EAMAttributeManagedBean.existingAttribute(),
EAMAttributeManagedBean.getAttrCount(),
EAMAttributeManagedBean.getCurrentAttributes(),
EAMAttributeManagedBean.newAttribute() und
EAMAttributeManagedBean.newAttributeType().

javax.faces.component.html.HtmlInputHidden attributeCounter [private]

Definiert in Zeile 35 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.checkDeleteAttribute(),
EAMAttributeManagedBean.editAttribute(),
EAMAttributeManagedBean.getAttributeCounter() und
EAMAttributeManagedBean.setAttributeCounter().

javax.faces.component.html.HtmlInputHidden attributeCounter2 [private]

Definiert in Zeile 36 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.addExistingAttribute(),
EAMAttributeManagedBean.getAttributeCounter2() und
EAMAttributeManagedBean.setAttributeCounter2().

javax.faces.component.html.HtmlInputHidden attributeDataTypeCounter [private]

Definiert in Zeile 38 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.addAttributeDataType(),
EAMAttributeManagedBean.getAttributeDataTypeCounter() und
EAMAttributeManagedBean.setAttributeDataTypeCounter().

javax.faces.component.html.HtmlInputHidden attributeTypeCounter [private]

Definiert in Zeile 37 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.addAttributeType(),
EAMAttributeManagedBean.checkDeleteType(),
EAMAttributeManagedBean.editAttributeType(),
EAMAttributeManagedBean.getAttributeTypeCounter() und
EAMAttributeManagedBean.setAttributeTypeCounter().

boolean attributeTypeSelected [private]

Definiert in Zeile 48 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.backToAttribute(),
EAMAttributeManagedBean.cancelTempAttribute(),
EAMAttributeManagedBean.checkDeleteAttribute(),
EAMAttributeManagedBean.checkDeleteType(),
EAMAttributeManagedBean.editAttribute(),
EAMAttributeManagedBean.existingAttribute(),
EAMAttributeManagedBean.isAttributeTypeSelected(),
EAMAttributeManagedBean.newAttribute() und
EAMAttributeManagedBean.saveAttribute().

long checkDelete [private]

Definiert in Zeile 44 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.checkDeleteAttribute(),
EAMAttributeManagedBean.checkDeleteType(),
EAMAttributeManagedBean.deleteAttribute(),
EAMAttributeManagedBean.deleteAttributeType() und

EAMAttributeManagedBean.getCheckDelete().

boolean checkDeleteSelected [private]

Definiert in Zeile 51 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.checkDeleteAttribute(),
EAMAttributeManagedBean.checkDeleteType(),
EAMAttributeManagedBean.deleteAttribute(),
EAMAttributeManagedBean.deleteAttributeType(),
EAMAttributeManagedBean.editAttribute(),
EAMAttributeManagedBean.editAttributeType(),
EAMAttributeManagedBean.existingAttribute(),
EAMAttributeManagedBean.isCheckDeleteSelected(),
EAMAttributeManagedBean.newAttribute(),
EAMAttributeManagedBean.newAttributeType() und
EAMAttributeManagedBean.noDelete().

boolean createExAttribute = true [private]

Definiert in Zeile 50 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.backToAttribute(),
EAMAttributeManagedBean.cancelTempAttribute(),
EAMAttributeManagedBean.checkDeleteAttribute(),
EAMAttributeManagedBean.deleteAttribute(),
EAMAttributeManagedBean.editAttribute(),
EAMAttributeManagedBean.existingAttribute(),
EAMAttributeManagedBean.isCreateExAttribute(),
EAMAttributeManagedBean.newAttribute(), EAMAttributeManagedBean.noDelete() und
EAMAttributeManagedBean.saveAttribute().

boolean createNewAttribute = true [private]

Definiert in Zeile 52 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.backToAttribute(),
EAMAttributeManagedBean.cancelTempAttribute(),
EAMAttributeManagedBean.checkDeleteAttribute(),
EAMAttributeManagedBean.deleteAttribute(),
EAMAttributeManagedBean.editAttribute(),
EAMAttributeManagedBean.existingAttribute(),
EAMAttributeManagedBean.isCreateNewAttribute(),
EAMAttributeManagedBean.newAttribute(), EAMAttributeManagedBean.noDelete() und
EAMAttributeManagedBean.saveAttribute().

boolean createNewAttributeType = true [private]

Definiert in Zeile 53 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.backToAttributeType(),
EAMAttributeManagedBean.checkDeleteType(),
EAMAttributeManagedBean.deleteAttributeType(),
EAMAttributeManagedBean.editAttributeType(),
EAMAttributeManagedBean.isCreateNewAttributeType(),
EAMAttributeManagedBean.newAttributeType(),
EAMAttributeManagedBean.noDelete() und
EAMAttributeManagedBean.saveAttributeType().

ArrayList<EAMAttribute> currentAttributes = new ArrayList<EAMAttribute>()

Definiert in Zeile 33 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.getCurrentAttributes().

EAMAttributeType currentAttributeName [private]

Definiert in Zeile 42 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.getCurrentAttributeName().

boolean exAttributeSelected [private]

Definiert in Zeile 49 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.cancelTempAttribute(),
EAMAttributeManagedBean.checkDeleteAttribute(),
EAMAttributeManagedBean.editAttribute(),
EAMAttributeManagedBean.existingAttribute(),
EAMAttributeManagedBean.isExAttributeSelected(),
EAMAttributeManagedBean.newAttribute() und
EAMAttributeManagedBean.saveAttribute().

ArrayList<EAMAttribute> existingAttributes = new ArrayList<EAMAttribute>()

Definiert in Zeile 34 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.addExistingAttribute() und
EAMAttributeManagedBean.getExistingAttributes().

**EAMStaticClass factory = EAMStaticClass.getInstance(userId, password,
this.getClass().getName()) [private]**

Definiert in Zeile 30 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.checkDeleteAttribute(),
EAMAttributeManagedBean.checkDeleteType(),

EAMAttributeManagedBean.deleteAttribute(),
EAMAttributeManagedBean.deleteAttributeType(),
EAMAttributeManagedBean.editAttribute(),
EAMAttributeManagedBean.editAttributeType(),
EAMAttributeManagedBean.getCurrentAttributes(),
EAMAttributeManagedBean.getExistingAttributes(),
EAMAttributeManagedBean.getPossibleAttributeTypes(),
EAMAttributeManagedBean.saveAttribute() und
EAMAttributeManagedBean.saveAttributeType().

String password =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")

Definiert in Zeile 28 der Datei EAMAttributeManagedBean.java.

ArrayList<DataType> possibleAttributeDataTypes = new ArrayList<DataType>()
[private]

Definiert in Zeile 43 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.addAttributeDataType() und
EAMAttributeManagedBean.getPossibleAttributeDataTypes().

ArrayList<EAMAttributeType> possibleAttributeTypes = new
ArrayList<EAMAttributeType>() [private]

Definiert in Zeile 40 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.addAttributeType() und
EAMAttributeManagedBean.getPossibleAttributeTypes().

int saveCounter = 0 [private]

Definiert in Zeile 45 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.saveAttribute().

boolean sortAscendingOver = true [private]

Definiert in Zeile 57 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.isSortAscendingOver().

String sortColumnOver = null [private]

Definiert in Zeile 56 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.getSortColumnOver().

EAMAttribute tempAttribute [private]

Definiert in Zeile 39 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.addAttributeType(),
EAMAttributeManagedBean.addExistingAttribute(),
EAMAttributeManagedBean.checkDeleteAttribute(),
EAMAttributeManagedBean.editAttribute(),
EAMAttributeManagedBean.existingAttribute(),
EAMAttributeManagedBean.getTempAttribute(),
EAMAttributeManagedBean.newAttribute() und
EAMAttributeManagedBean.saveAttribute().

boolean tempAttributeSelected [private]

Definiert in Zeile 47 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.backToAttribute(),
EAMAttributeManagedBean.cancelTempAttribute(),
EAMAttributeManagedBean.checkDeleteAttribute(),
EAMAttributeManagedBean.editAttribute(),
EAMAttributeManagedBean.existingAttribute(),
EAMAttributeManagedBean.isTempAttributeSelected(),
EAMAttributeManagedBean.newAttribute() und
EAMAttributeManagedBean.saveAttribute().

EAMAttributeType tempAttributeType [private]

Definiert in Zeile 41 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.addAttributeDataType(),
EAMAttributeManagedBean.checkDeleteType(),
EAMAttributeManagedBean.editAttributeType(),
EAMAttributeManagedBean.getTempAttributeType(),
EAMAttributeManagedBean.newAttributeType() und
EAMAttributeManagedBean.saveAttributeType().

boolean tempAttributeTypeSelected [private]

Definiert in Zeile 46 der Datei EAMAttributeManagedBean.java.

Wird benutzt von EAMAttributeManagedBean.backToAttributeType(),
EAMAttributeManagedBean.checkDeleteType(),
EAMAttributeManagedBean.editAttributeType(),
EAMAttributeManagedBean.isTempAttributeTypeSelected(),
EAMAttributeManagedBean.newAttributeType() und
EAMAttributeManagedBean.saveAttributeType().

```
String userId =  
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")
```

Definiert in Zeile 27 der Datei EAMAttributeManagedBean.java.

4.4.2 EAMCategoryManagedBean Klassenreferenz

4.4.2.1 Ausführliche Beschreibung

ManagedBean für alle Aktionen mit Metamodellen

Autor:

David Heimann
Hyung-Bin Kim

Definiert in Zeile 18 der Datei EAMCategoryManagedBean.java.

Öffentliche Methoden

- `String back ()`
- `String checkDelete ()`
- `String deleteCategory ()` throws Exception
- `String editCategory ()`
- `javax.faces.component.html.HtmlInputHidden getCategoryCounter ()`
- `long getCheckDelete ()`
- `ArrayList< EAMCategory > getCurrentCategories ()`
- `ArrayList< EAMCategory > getCurrentEAMCategories ()`
- `javax.faces.component.html.HtmlInputHidden getRowCounter ()`
- `EAMCategory getTempCategory ()`
- `boolean isCheckDeleteSelected ()`
- `boolean isCreateNewCategory ()`
- `boolean isEditCategorySelected ()`
- `boolean isNewCategorySelected ()`
- `boolean isNewCreatedCategory ()`
- `String newCategory ()`
- `String noDelete ()`
- `String openCategory ()`
- `String saveCategory ()` throws Exception
- `void setCategoryCounter (javax.faces.component.html.HtmlInputHidden categoryCounter)`
- `void setCheckDelete (long checkDelete)`

- void setCheckDeleteSelected (boolean checkDeleteSelected)
- void setCreateNewCategory (boolean createNewCategory)
- void setCurrentCategories (ArrayList< EAMCategory > currentCategories)
- void setCurrentEAMCategories (ArrayList< EAMCategory > currentEAMCategories)
- void setEditCategorySelected (boolean editCategorySelected)
- void setNewCategorySelected (boolean newCategorySelected)
- void setNewCreatedCategory (boolean newCreatedCategory)
- void setRowCounter (javax.faces.component.html.HtmlInputHidden rowCounter)
- void setTempCategory (EAMCategory tempCategory)

Öffentliche Attribute

- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Private Attribute

- javax.faces.component.html.HtmlInputHidden categoryCounter
- long checkDelete
- boolean checkDeleteSelected
- boolean createNewCategory = true
- ArrayList< EAMCategory > currentCategories = new ArrayList<EAMCategory>()
- boolean editCategorySelected
- EAMStaticClass factory = EAMStaticClass.getInstance(userId, password, this.getClass().getName())
- boolean newCategorySelected
- boolean newCreatedCategory = false
- EAMCategory tempCategory

4.4.2.2 Dokumentation der Elementfunktionen

String back ()

Definiert in Zeile 231 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.createNewCategory,
EAMCategoryManagedBean.editCategorySelected,
EAMCategoryManagedBean.newCategorySelected und
EAMCategoryManagedBean.tempCategory.

String checkDelete ()

Definiert in Zeile 197 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.categoryCounter,
EAMStaticClass.categoryDAOInterface, EAMCategoryManagedBean.checkDeleteSelected,
EAMCategoryManagedBean.createNewCategory,
EAMCategoryManagedBean.editCategorySelected, EAMCategoryManagedBean.factory,
EAMCategoryManagedBean.newCategorySelected und
EAMCategoryManagedBean.tempCategory.

Wird benutzt von EAMCategoryManagedBean.deleteCategory() und
EAMCategoryManagedBean.getCheckDelete().

String deleteCategory () throws Exception

Definiert in Zeile 211 der Datei EAMCategoryManagedBean.java.

Benutzt EAMStaticClass.categoryDAOInterface,
EAMCategoryManagedBean.checkDelete(),
EAMCategoryManagedBean.checkDeleteSelected,
EAMCategoryManagedBean.createNewCategory,
EAMCategoryManagedBean.currentCategories und EAMCategoryManagedBean.factory.

String editCategory ()

Holt die Informationen des Metamodells aus der DAO und stellt diese dar.

Rückgabe:

Definiert in Zeile 101 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.categoryCounter,
EAMStaticClass.categoryDAOInterface, EAMCategoryManagedBean.checkDeleteSelected,
EAMCategoryManagedBean.createNewCategory,
EAMCategoryManagedBean.editCategorySelected, EAMCategoryManagedBean.factory,
EAMCategoryManagedBean.newCategorySelected,
EAMCategoryManagedBean.newCreatedCategory und
EAMCategoryManagedBean.tempCategory.

javax.faces.component.html.HtmlInputHidden getCategoryCounter ()

Definiert in Zeile 265 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.categoryCounter.

long getCheckDelete ()

Definiert in Zeile 257 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.checkDelete().

ArrayList<EAMCategory> getCurrentCategories ()

Definiert in Zeile 282 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.currentCategories.

ArrayList<EAMCategory> getCurrentEAMCategories ()

Listet alle bereits vorhandenen Metamodelle auf.

Rückgabe:

Definiert in Zeile 44 der Datei EAMCategoryManagedBean.java.

Benutzt EAMStaticClass.categoryDAOInterface,

EAMCategoryManagedBean.currentCategories und EAMCategoryManagedBean.factory.

javax.faces.component.html.HtmlInputHidden getRowCounter ()

Definiert in Zeile 243 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.categoryCounter.

EAMCategory getTempCategory ()

Definiert in Zeile 274 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.tempCategory.

boolean isCheckDeleteSelected ()

Definiert in Zeile 306 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.checkDeleteSelected.

boolean isCreateNewCategory ()

Definiert in Zeile 314 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.createNewCategory.

boolean isEditCategorySelected ()

Definiert in Zeile 298 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.editCategorySelected.

boolean isNewCategorySelected ()

Definiert in Zeile 290 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.newCategorySelected.

boolean isNewCreatedCategory ()

Definiert in Zeile 322 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.newCreatedCategory.

String newCategory ()

Definiert in Zeile 85 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.checkDeleteSelected,
EAMCategoryManagedBean.createNewCategory,
EAMCategoryManagedBean.editCategorySelected,
EAMCategoryManagedBean.newCategorySelected und
EAMCategoryManagedBean.tempCategory.

String noDelete ()

Definiert in Zeile 222 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.checkDeleteSelected und
EAMCategoryManagedBean.createNewCategory.

String openCategory ()

Navigiert zum ausgewählten Metamodell.

Rückgabe:

Definiert in Zeile 64 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.categoryCounter,
EAMCategoryManagedBean.checkDeleteSelected,
EAMCategoryManagedBean.currentCategories, EAMStaticClass.currentCategory,
EAMCategoryManagedBean.editCategorySelected, EAMCategoryManagedBean.factory,
EAMCategoryManagedBean.newCategorySelected und
EAMCategoryManagedBean.newCreatedCategory.

String saveCategory () throws Exception

Überprüfung der Korrektheit der Eingaben und Speichern des Metamodells in die DAO.

Rückgabe:

Ausnahmebehandlung:

Exception

Definiert in Zeile 123 der Datei EAMCategoryManagedBean.java.

Benutzt EAMStaticClass.categoryDAOInterface,
EAMCategoryManagedBean.createNewCategory,
EAMCategoryManagedBean.currentCategories,
EAMCategoryManagedBean.editCategorySelected, EAMCategoryManagedBean.factory,
EAMCategoryManagedBean.newCategorySelected,
EAMCategoryManagedBean.newCreatedCategory und
EAMCategoryManagedBean.tempCategory.

**void setCategoryCounter (javax.faces.component.html.HtmlInputHidden
categoryCounter)**

Definiert in Zeile 269 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.categoryCounter.

void setCheckDelete (long checkDelete)

Definiert in Zeile 261 der Datei EAMCategoryManagedBean.java.

void setCheckDeleteSelected (boolean checkDeleteSelected)

Definiert in Zeile 310 der Datei EAMCategoryManagedBean.java.

void setCreateNewCategory (boolean createNewCategory)

Definiert in Zeile 318 der Datei EAMCategoryManagedBean.java.

void setCurrentCategories (ArrayList< EAMCategory > currentCategories)

Definiert in Zeile 286 der Datei EAMCategoryManagedBean.java.

void setCurrentEAMCategories (ArrayList< EAMCategory > currentEAMCategories)

Definiert in Zeile 252 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.currentCategories.

void setEditCategorySelected (boolean editCategorySelected)

Definiert in Zeile 302 der Datei EAMCategoryManagedBean.java.

void setNewCategorySelected (boolean newCategorySelected)

Definiert in Zeile 294 der Datei EAMCategoryManagedBean.java.

void setNewCreatedCategory (boolean newCreatedCategory)

Definiert in Zeile 326 der Datei EAMCategoryManagedBean.java.

void setRowCounter (javax.faces.component.html.HtmlInputHidden rowCounter)

Definiert in Zeile 248 der Datei EAMCategoryManagedBean.java.

Benutzt EAMCategoryManagedBean.categoryCounter.

void setTempCategory (EAMCategory tempCategory)

Definiert in Zeile 278 der Datei EAMCategoryManagedBean.java.

4.4.2.3 Dokumentation der Datenelemente

javax.faces.component.html.HtmlInputHidden categoryCounter [private]

Definiert in Zeile 28 der Datei EAMCategoryManagedBean.java.

Wird benutzt von EAMCategoryManagedBean.checkDelete(),
EAMCategoryManagedBean.editCategory(),
EAMCategoryManagedBean.getCategoryCounter(),
EAMCategoryManagedBean.getRowCounter(),
EAMCategoryManagedBean.openCategory(),
EAMCategoryManagedBean.setCategoryCounter() und
EAMCategoryManagedBean.setRowCounter().

long checkDelete [private]

Definiert in Zeile 32 der Datei EAMCategoryManagedBean.java.

boolean checkDeleteSelected [private]

Definiert in Zeile 35 der Datei EAMCategoryManagedBean.java.

Wird benutzt von EAMCategoryManagedBean.checkDelete(),
EAMCategoryManagedBean.deleteCategory(),
EAMCategoryManagedBean.editCategory(),
EAMCategoryManagedBean.isCheckDeleteSelected(),

EAMCategoryManagedBean.newCategory(), EAMCategoryManagedBean.noDelete() und EAMCategoryManagedBean.openCategory().

boolean createNewCategory = true [private]

Definiert in Zeile 36 der Datei EAMCategoryManagedBean.java.

Wird benutzt von EAMCategoryManagedBean.back(), EAMCategoryManagedBean.checkDelete(), EAMCategoryManagedBean.deleteCategory(), EAMCategoryManagedBean.editCategory(), EAMCategoryManagedBean.isCreateNewCategory(), EAMCategoryManagedBean.newCategory(), EAMCategoryManagedBean.noDelete() und EAMCategoryManagedBean.saveCategory().

ArrayList<EAMCategory> currentCategories = new ArrayList<EAMCategory>() [private]

Definiert in Zeile 31 der Datei EAMCategoryManagedBean.java.

Wird benutzt von EAMCategoryManagedBean.deleteCategory(), EAMCategoryManagedBean.getCurrentCategories(), EAMCategoryManagedBean.getCurrentEAMCategories(), EAMCategoryManagedBean.openCategory(), EAMCategoryManagedBean.saveCategory() und EAMCategoryManagedBean.setCurrentEAMCategories().

boolean editCategorySelected [private]

Definiert in Zeile 34 der Datei EAMCategoryManagedBean.java.

Wird benutzt von EAMCategoryManagedBean.back(), EAMCategoryManagedBean.checkDelete(), EAMCategoryManagedBean.editCategory(), EAMCategoryManagedBean.isEditCategorySelected(), EAMCategoryManagedBean.newCategory(), EAMCategoryManagedBean.openCategory() und EAMCategoryManagedBean.saveCategory().

EAMStaticClass factory = EAMStaticClass.getInstance(userId, password, this.getClass().getName()) [private]

Definiert in Zeile 24 der Datei EAMCategoryManagedBean.java.

Wird benutzt von EAMCategoryManagedBean.checkDelete(), EAMCategoryManagedBean.deleteCategory(), EAMCategoryManagedBean.editCategory(), EAMCategoryManagedBean.getCurrentEAMCategories(), EAMCategoryManagedBean.openCategory() und EAMCategoryManagedBean.saveCategory().

boolean newCategorySelected [private]

Definiert in Zeile 33 der Datei EAMCategoryManagedBean.java.

Wird benutzt von EAMCategoryManagedBean.back(),
EAMCategoryManagedBean.checkDelete(), EAMCategoryManagedBean.editCategory(),
EAMCategoryManagedBean.isNewCategorySelected(),
EAMCategoryManagedBean.newCategory(), EAMCategoryManagedBean.openCategory()
und EAMCategoryManagedBean.saveCategory().

boolean newCreatedCategory = false [private]

Definiert in Zeile 37 der Datei EAMCategoryManagedBean.java.

Wird benutzt von EAMCategoryManagedBean.editCategory(),
EAMCategoryManagedBean.isNewCreatedCategory(),
EAMCategoryManagedBean.openCategory() und
EAMCategoryManagedBean.saveCategory().

String password =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")

Definiert in Zeile 22 der Datei EAMCategoryManagedBean.java.

EAMCategory tempCategory [private]

Definiert in Zeile 30 der Datei EAMCategoryManagedBean.java.

Wird benutzt von EAMCategoryManagedBean.back(),
EAMCategoryManagedBean.checkDelete(), EAMCategoryManagedBean.editCategory(),
EAMCategoryManagedBean.getTempCategory(),
EAMCategoryManagedBean.newCategory() und
EAMCategoryManagedBean.saveCategory().

String userId =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Definiert in Zeile 21 der Datei EAMCategoryManagedBean.java.

4.4.3 EAMObjectRelationManagedBean Klassenreferenz

4.4.3.1 Ausführliche Beschreibung

Managedbean für alle Aktionen mit Objekten und Relationen

Autor:

David Heimann
Hyung-Bin Kim

Definiert in Zeile 34 der Datei EAMObjectRelationManagedBean.java.

Öffentliche Methoden

- String addFirstRelationMember ()
- String addSecondRelationMember ()
- void addStatus ()
- String addType ()
- String backToObject ()
- String backToObjects ()
- String checkDeleteObject ()
- String checkDeleteRelation ()
- String checkDeleteStatus ()
- String checkDeleteTime ()
- String checkDeleteType ()
- String deleteObjectRelation () throws Exception
- String deleteStatus () throws Exception
- String deleteTime () throws Exception
- String deleteType () throws Exception
- String editObject ()
- String editRelation ()
- String editStatus ()
- String editTime ()
- String editType ()
- Calendar getCal ()
- Calendar getCal2 ()
- long getCheckDelete ()
- EAMSuperClass getCurrentObjectRelation ()
- ArrayList< EAMObject > getCurrentObjects ()
- ArrayList< EAMRelation > getCurrentRelations ()
- String getDay ()
- String getDay2 ()
- String getErrorMsg ()
- javax.faces.component.html.HtmlInputHidden
getFirstRelationMemberFromOtherMetamodelsCounter ()
- javax.faces.component.html.HtmlInputHidden
getFirstRelationMemberFromOwnMetamodelCounter ()
- String getFirstRelationMemberMultiplicity ()
- String getMonth ()

- String getMonth2 ()
- EAMStatus getNewStatus ()
- javax.faces.component.html.HtmlInputHidden getObjectCounter ()
- ArrayList< EAMObject > getPossibleRelationsFromOtherMetamodels ()
- ArrayList< EAMObject > getPossibleRelationsFromOwnMetamodel ()
- ArrayList< EAMStatus > getPossibleStatuses () throws InstantiationException, IllegalAccessException
- ArrayList< EAMType > getPossibleTypes ()
- javax.faces.component.html.HtmlInputHidden getRelationCounter ()
- javax.faces.component.html.HtmlInputHidden getRelationMemberCounter ()
- javax.faces.component.html.HtmlInputHidden getRowCount ()
- javax.faces.component.html.HtmlInputHidden getSecondRelationMemberFromOtherMetamodelsCounter ()
- javax.faces.component.html.HtmlInputHidden getSecondRelationMemberFromOwnMetamodelCounter ()
- String getSecondRelationMemberMultiplicity ()
- String getSortColumnOver ()
- int getStatusCount ()
- javax.faces.component.html.HtmlInputHidden getStatusCounter ()
- EAMSuperClass getTempObjectRelation ()
- EAMPeriodTime getTempPeriodTime ()
- EAMPointTime getTempPointTime ()
- EAMStatus getTempStatus ()
- EAMTime getTempTime ()
- EAMType getTempType ()
- int getTypeCount ()
- javax.faces.component.html.HtmlInputHidden getTypeCounter ()
- String getYear ()
- String getYear2 ()
- boolean isCheckDeleteSelected ()
- boolean isCheckDeleteStatus ()
- boolean isCheckDeleteTime ()
- boolean isCheckDeleteType ()
- boolean isCreateNewObject ()
- boolean isCreateNewRelation ()
- boolean isCreateNewStatus ()
- boolean isCreateNewTime ()
- boolean isCreateNewType ()
- boolean isEditObjectTypeSelected ()

- boolean isEditStatusSelected ()
- boolean isEditTime ()
- boolean isEditTypeSelected ()
- boolean isFirstObjectOtherSelected ()
- boolean isNewObjectTypeSelected ()
- boolean isNewPeriodTimeSelected ()
- boolean isNewPointTimeSelected ()
- boolean isNewStatusSelected ()
- boolean isNewTypeSelected ()
- boolean isObjectType ()
- boolean isPeriodTime ()
- boolean isRelationType ()
- boolean isShowCurrentTime ()
- boolean isShowCurrentTimeButton ()
- boolean isShowPeriodTime ()
- boolean isShowPointTime ()
- boolean isSortAscendingOver ()
- String newObject ()
- String newPeriodTime ()
- String newPointTime ()
- String newRelation ()
- String newStatus ()
- String newType ()
- String noDelete ()
- String saveObjectRelation () throws Exception
- String saveStatus () throws Exception
- String saveTime ()
- String saveType () throws Exception
- void setCal (Calendar cal)
- void setCal2 (Calendar cal2)
- void setCheckDelete (long checkDelete)
- void setCheckDeleteSelected (boolean checkDeleteSelected)
- void setCheckDeleteStatus (boolean checkDeleteStatus)
- void setCheckDeleteTime (boolean checkDeleteTime)
- void setCheckDeleteType (boolean checkDeleteType)
- void setCreateNewObject (boolean createNewObject)
- void setCreateNewRelation (boolean createNewRelation)
- void setCreateNewStatus (boolean createNewStatus)

- void setCreateNewTime (boolean createNewTime)
- void setCreateNewType (boolean createNewType)
- void setCurrentObjectRelation (EAMSuperClass currentObjectRelation)
- void setCurrentObjects (ArrayList< EAMObject > currentObjects)
- void setCurrentRelations (ArrayList< EAMRelation > currentRelations)
- void setDay (String day)
- void setDay2 (String day2)
- void setEditObjectTypeSelected (boolean editObjectTypeSelected)
- void setEditStatusSelected (boolean editStatusSelected)
- void setEditTime (boolean editTime)
- void setEditTypeSelected (boolean editTypeSelected)
- void setErrorMsg (String errorMsg)
- void setFirstObjectOtherSelected (boolean firstObjectOtherSelected)
- void setFirstRelationMemberFromOtherMetamodelsCounter
(javax.faces.component.html.HtmlInputHidden
firstRelationMemberFromOtherMetamodelsCounter)
- void setFirstRelationMemberFromOwnMetamodelCounter
(javax.faces.component.html.HtmlInputHidden
firstRelationMemberFromOwnMetamodelCounter)
- void setFirstRelationMemberMultiplicity (String firstRelationMemberMultiplicity)
- void setMonth (String month)
- void setMonth2 (String month2)
- void setNewObjectTypeSelected (boolean newObjectTypeSelected)
- void setNewPeriodTimeSelected (boolean newPeriodTimeSelected)
- void setNewPointTimeSelected (boolean newPointTimeSelected)
- void setNewStatus (EAMStatus newStatus)
- void setNewStatusSelected (boolean newStatusSelected)
- void setNewTypeSelected (boolean newTypeSelected)
- void setObjectCounter (javax.faces.component.html.HtmlInputHidden
objectCounter)
- void setObjectType (boolean objectType)
- void setPeriodTime (boolean periodTime)
- void setPossibleRelationsFromOtherMetamodels (ArrayList< EAMObject >
possibleRelationsFromOtherMetamodels)
- void setPossibleRelationsFromOwnMetamodel (ArrayList< EAMObject >
possibleRelationsFromOwnMetamodel)
- void setPossibleStatuses (ArrayList< EAMStatus > possibleStatuses)
- void setPossibleTypes (ArrayList< EAMType > possibleTypes)
- void setRelationCounter (javax.faces.component.html.HtmlInputHidden

- relationCounter)
- void setRelationMemberCounter (javax.faces.component.html.HtmlInputHidden relationMemberCounter)
- void setRelationType (boolean relationType)
- void setRowCounter (javax.faces.component.html.HtmlInputHidden rowCounter)
- void setSecondRelationMemberFromOtherMetamodelsCounter (javax.faces.component.html.HtmlInputHidden secondRelationMemberFromOtherMetamodelsCounter)
- void setSecondRelationMemberFromOwnMetamodelCounter (javax.faces.component.html.HtmlInputHidden secondRelationMemberFromOwnMetamodelCounter)
- void setSecondRelationMemberMultiplicity (String secondRelationMemberMultiplicity)
- void setShowCurrentTime (boolean showCurrentTime)
- void setShowCurrentTimeButton (boolean showCurrentTimeButton)
- void setShowPeriodTime (boolean showPeriodTime)
- void setShowPointTime (boolean showPointTime)
- void setSortAscendingOver (boolean sortAscendingOver)
- void setSortColumnOver (String sortColumnOver)
- void setStatusCount (int statusCount)
- void setStatusCounter (javax.faces.component.html.HtmlInputHidden statusCounter)
- void setTempObjectRelation (EAMSuperClass tempObjectRelation)
- void setTempPeriodTime (EAMPeriodTime tempPeriodTime)
- void setTempPointTime (EAMPointTime tempPointTime)
- void setTempStatus (EAMStatus tempStatus)
- void setTempTime (EAMTime tempTime)
- void setTempType (EAMType tempType)
- void setTypeCount (int typeCount)
- void setTypeCounter (javax.faces.component.html.HtmlInputHidden typeCounter)
- void setYear (String year)
- void setYear2 (String year2)

Öffentliche Attribute

- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Private Attribute

- Calendar cal
- Calendar cal2
- long checkDelete
- boolean checkDeleteSelected
- boolean checkDeleteStatus = false
- boolean checkDeleteTime = false
- boolean checkDeleteType = false
- boolean createNewObject = true
- boolean createNewRelation = true
- boolean createNewStatus = true
- boolean createNewTime = true
- boolean createNewType = true
- EAMSuperClass currentObjectRelation
- ArrayList< EAMObject > currentObjects = new ArrayList<EAMObject>()
- ArrayList< EAMRelation > currentRelations = new ArrayList<EAMRelation>()
- EAMType currentType
- String day
- String day2
- boolean editStatusSelected = false
- boolean editTime = false
- boolean editTypeSelected = false
- String errorMsg
- EAMStaticClass factory = EAMStaticClass.getInstance(userId, password, this.getClass().getName())
- boolean firstObjectOtherSelected = true
- javax.faces.component.html.HtmlInputHidden
firstRelationMemberFromOtherMetamodelsCounter
- javax.faces.component.html.HtmlInputHidden
firstRelationMemberFromOwnMetamodelCounter
- String firstRelationMemberMultiplicity
- String month
- String month2
- boolean newPeriodTimeSelected = false
- boolean newPointTimeSelected = false
- boolean newStatusSelected = false
- boolean newTypeSelected = false
- javax.faces.component.html.HtmlInputHidden objectCounter

- boolean objectType
- boolean periodTime
- ArrayList< EAMObject > possibleRelationsFromOtherMetamodels = new ArrayList<EAMObject>()
- ArrayList< EAMObject > possibleRelationsFromOwnMetamodel = new ArrayList<EAMObject>()
- ArrayList< EAMStatus > possibleStatuses = new ArrayList<EAMStatus>()
- ArrayList< EAMType > possibleTypes = new ArrayList<EAMType>()
- javax.faces.component.html.HtmlInputHidden relationCounter
- boolean relationType
- javax.faces.component.html.HtmlInputHidden
secondRelationMemberFromOtherMetamodelsCounter
- javax.faces.component.html.HtmlInputHidden
secondRelationMemberFromOwnMetamodelCounter
- String secondRelationMemberMultiplicity
- boolean showCurrentTime = false
- boolean showCurrentTimeButton = false
- boolean showPeriodTime = false
- boolean showPointTime = false
- boolean sortAscendingOver = true
- String sortColumnOver = null
- int statusCount = 0
- javax.faces.component.html.HtmlInputHidden statusCounter
- EAMSuperClass tempObjectRelation
- EAMPeriodTime tempPeriodTime
- EAMPointTime tempPointTime
- EAMStatus tempStatus
- EAMTime tempTime
- EAMType tempType
- int typeCount = 0
- javax.faces.component.html.HtmlInputHidden typeCounter
- String year
- String year2

4.4.3.2 Dokumentation der Elementfunktionen

String addFirstRelationMember ()

Fügt entweder ein Objekt aus dem eigenen Metamodell oder aus den anderen Metamodellen als erstes Relationsobjekt hinzu.

Rückgabe:

Definiert in Zeile 1010 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.firstObjectOtherSelected,
EAMObjectRelationManagedBean.firstRelationMemberFromOtherMetamodelsCounter,
EAMObjectRelationManagedBean.firstRelationMemberFromOwnMetamodelCounter,
EAMObjectRelationManagedBean.possibleRelationsFromOtherMetamodels,
EAMObjectRelationManagedBean.possibleRelationsFromOwnMetamodel und
EAMObjectRelationManagedBean.tempObjectRelation.

String addSecondRelationMember ()

Fügt entweder ein Objekt aus dem eigenen Metamodell oder aus den anderen Metamodellen als zweites Relationsobjekt hinzu.

Rückgabe:

Definiert in Zeile 1064 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.possibleRelationsFromOtherMetamodels,
EAMObjectRelationManagedBean.possibleRelationsFromOwnMetamodel,
EAMObjectRelationManagedBean.secondRelationMemberFromOtherMetamodelsCounter,
EAMObjectRelationManagedBean.secondRelationMemberFromOwnMetamodelCounter
und EAMObjectRelationManagedBean.tempObjectRelation.

void addStatus ()

Fügt dem Objekt einen Status hinzu.

Definiert in Zeile 796 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.possibleStatuses,
EAMObjectRelationManagedBean.statusCounter und
EAMObjectRelationManagedBean.tempObjectRelation.

String addType ()

Fügt entweder ein Objekttyp oder ein Relationstyp hinzu.

Rückgabe:

Definiert in Zeile 1141 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.possibleTypes,
EAMObjectRelationManagedBean.tempObjectRelation und
EAMObjectRelationManagedBean.typeCounter.

String backToObject ()

Setzt alle booleans bei der Objekt/Relation JSF-Seite zurück.

Definiert in Zeile 1731 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteSelected,
EAMObjectRelationManagedBean.checkDeleteType(),
EAMObjectRelationManagedBean.createNewObject,
EAMObjectRelationManagedBean.createNewRelation,
EAMObjectRelationManagedBean.createNewStatus,
EAMObjectRelationManagedBean.createNewTime,
EAMObjectRelationManagedBean.createNewType,
EAMObjectRelationManagedBean.editStatusSelected,
EAMObjectRelationManagedBean.editTypeSelected,
EAMObjectRelationManagedBean.newPeriodTimeSelected,
EAMObjectRelationManagedBean.newPointTimeSelected,
EAMObjectRelationManagedBean.newStatusSelected,
EAMObjectRelationManagedBean.newTypeSelected,
EAMObjectRelationManagedBean.showCurrentTime,
EAMObjectRelationManagedBean.showCurrentTimeButton,
EAMObjectRelationManagedBean.showPeriodTime,
EAMObjectRelationManagedBean.showPointTime und
EAMObjectRelationManagedBean.tempObjectRelation.

Wird benutzt von EAMObjectRelationManagedBean.getPossibleStatuses() und
EAMObjectRelationManagedBean.getPossibleTypes().

String backToObjects ()

Setzt alle booleans bei der Objekt/Relationsübersicht JSF Seite zurück.

Rückgabe:

Definiert in Zeile 1768 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.createNewObject,
EAMObjectRelationManagedBean.createNewRelation,
EAMObjectRelationManagedBean.currentObjects und
EAMObjectRelationManagedBean.getCurrentObjects().

String checkDeleteObject ()

Hinweis ob das angeklickte Objekt gelöscht werden soll.

Definiert in Zeile 617 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteSelected,
EAMObjectRelationManagedBean.createNewObject,

EAMObjectRelationManagedBean.currentObjectRelation,
EAMObjectRelationManagedBean.factory,
EAMObjectRelationManagedBean.objectCounter, EAMStaticClass.objectDAOInterface
und EAMObjectRelationManagedBean.tempObjectRelation.

String checkDeleteRelation ()

Hinweis ob die angeklickte Relation gelöscht werden soll.

Definiert in Zeile 633 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteSelected,
EAMObjectRelationManagedBean.createNewRelation,
EAMObjectRelationManagedBean.currentObjectRelation,
EAMObjectRelationManagedBean.factory,
EAMObjectRelationManagedBean.relationCounter,
EAMStaticClass.relationDAOInterface und
EAMObjectRelationManagedBean.tempObjectRelation.

String checkDeleteStatus ()

Definiert in Zeile 665 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDelete,
EAMObjectRelationManagedBean.checkDeleteSelected,
EAMObjectRelationManagedBean.createNewStatus,
EAMObjectRelationManagedBean.editStatusSelected,
EAMObjectRelationManagedBean.factory,
EAMObjectRelationManagedBean.newStatusSelected,
EAMObjectRelationManagedBean.statusCount,
EAMObjectRelationManagedBean.statusCounter, EAMStaticClass.statusDAOInterface
und EAMObjectRelationManagedBean.tempStatus.

Wird benutzt von EAMObjectRelationManagedBean.deleteStatus(),
EAMObjectRelationManagedBean.isCheckDeleteStatus() und
EAMObjectRelationManagedBean.noDelete().

String checkDeleteTime ()

Definiert in Zeile 697 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDelete,
EAMObjectRelationManagedBean.createNewTime,
EAMObjectRelationManagedBean.factory,
EAMObjectRelationManagedBean.showCurrentTimeButton,
EAMObjectRelationManagedBean.tempObjectRelation,
EAMObjectRelationManagedBean.tempTime und EAMStaticClass.timeDAOInterface.

Wird benutzt von EAMObjectRelationManagedBean.deleteTime(),

EAMObjectRelationManagedBean.editObject(),
EAMObjectRelationManagedBean.editRelation(),
EAMObjectRelationManagedBean.isCheckDeleteTime() und
EAMObjectRelationManagedBean.noDelete().

String checkDeleteType ()

Definiert in Zeile 734 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.createNewType,
EAMObjectRelationManagedBean.currentType,
EAMObjectRelationManagedBean.editTypeSelected,
EAMObjectRelationManagedBean.factory, EAMObjectRelationManagedBean.objectType,
EAMStaticClass.objectTypeDAOInterface, EAMStaticClass.relationTypeDAOInterface,
EAMObjectRelationManagedBean.tempType,
EAMObjectRelationManagedBean.typeCount und
EAMObjectRelationManagedBean.typeCounter.

Wird benutzt von EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.deleteType(),
EAMObjectRelationManagedBean.isCheckDeleteType() und
EAMObjectRelationManagedBean.noDelete().

String deleteObjectRelation () throws Exception

Löscht das entweder ein EAMObject oder eine EAMRelation aus der DAO.

Rückgabe:

Ausnahmebehandlung:

Exception

Definiert in Zeile 649 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteSelected,
EAMObjectRelationManagedBean.createNewObject,
EAMObjectRelationManagedBean.createNewRelation,
EAMObjectRelationManagedBean.currentObjectRelation,
EAMObjectRelationManagedBean.factory, EAMStaticClass.objectDAOInterface und
EAMStaticClass.relationDAOInterface.

String deleteStatus () throws Exception

Definiert in Zeile 682 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDelete,
EAMObjectRelationManagedBean.checkDeleteSelected,
EAMObjectRelationManagedBean.checkDeleteStatus(),

EAMObjectRelationManagedBean.createNewStatus,
EAMObjectRelationManagedBean.factory,
EAMObjectRelationManagedBean.possibleStatuses und
EAMStaticClass.statusDAOInterface.

String deleteTime () throws Exception

Definiert in Zeile 715 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDelete,
EAMObjectRelationManagedBean.checkDeleteTime(),
EAMObjectRelationManagedBean.createNewTime,
EAMObjectRelationManagedBean.factory,
EAMObjectRelationManagedBean.showCurrentTime,
EAMObjectRelationManagedBean.showPeriodTime,
EAMObjectRelationManagedBean.showPointTime,
EAMObjectRelationManagedBean.tempObjectRelation und
EAMStaticClass.timeDAOInterface.

String deleteType () throws Exception

Definiert in Zeile 754 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteType(),
EAMObjectRelationManagedBean.createNewType,
EAMObjectRelationManagedBean.currentType,
EAMObjectRelationManagedBean.factory, EAMObjectRelationManagedBean.objectType,
EAMStaticClass.objectTypeDAOInterface,
EAMObjectRelationManagedBean.possibleTypes und
EAMStaticClass.relationTypeDAOInterface.

String editObject ()

Holt die Informationen über das zu editierende EAMObject aus der DAO und stellt diese bereit.

Rückgabe:

Definiert in Zeile 436 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteSelected,
EAMObjectRelationManagedBean.checkDeleteTime(),
EAMObjectRelationManagedBean.createNewTime,
EAMStaticClass.currentObjectRelation,
EAMObjectRelationManagedBean.editStatusSelected,
EAMObjectRelationManagedBean.editTypeSelected,
EAMObjectRelationManagedBean.factory,

EAMObjectRelationManagedBean.newPeriodTimeSelected,
EAMObjectRelationManagedBean.newPointTimeSelected,
EAMObjectRelationManagedBean.newStatusSelected,
EAMObjectRelationManagedBean.newTypeSelected,
EAMObjectRelationManagedBean.objectCounter, EAMStaticClass.objectDAOInterface,
EAMObjectRelationManagedBean.objectType,
EAMObjectRelationManagedBean.relationType,
EAMObjectRelationManagedBean.showCurrentTime,
EAMObjectRelationManagedBean.showCurrentTimeButton,
EAMObjectRelationManagedBean.showPeriodTime,
EAMObjectRelationManagedBean.showPointTime und
EAMObjectRelationManagedBean.tempObjectRelation.

String editRelation ()

Holt die Informationen über die zu editierende EAMRelation aus der DAO und stellt diese bereit.

Rückgabe:

Definiert in Zeile 490 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteSelected,
EAMObjectRelationManagedBean.checkDeleteTime(),
EAMObjectRelationManagedBean.createNewTime,
EAMStaticClass.currentObjectRelation,
EAMObjectRelationManagedBean.editStatusSelected,
EAMObjectRelationManagedBean.editTypeSelected,
EAMObjectRelationManagedBean.factory,
EAMObjectRelationManagedBean.firstRelationMemberMultiplicity,
EAMObjectRelationManagedBean.newPeriodTimeSelected,
EAMObjectRelationManagedBean.newPointTimeSelected,
EAMObjectRelationManagedBean.newStatusSelected,
EAMObjectRelationManagedBean.newTypeSelected,
EAMObjectRelationManagedBean.objectType,
EAMObjectRelationManagedBean.relationCounter,
EAMStaticClass.relationDAOInterface, EAMObjectRelationManagedBean.relationType,
EAMObjectRelationManagedBean.secondRelationMemberMultiplicity,
EAMObjectRelationManagedBean.showCurrentTime,
EAMObjectRelationManagedBean.showCurrentTimeButton,
EAMObjectRelationManagedBean.showPeriodTime,
EAMObjectRelationManagedBean.showPointTime und
EAMObjectRelationManagedBean.tempObjectRelation.

String editStatus ()

Holt die Informationen über den zu editierenden Status aus der DAO und stellt diese bereit.

Rückgabe:

Definiert in Zeile 899 der Datei `EAMObjectRelationManagedBean.java`.

Benutzt `EAMObjectRelationManagedBean.checkDeleteSelected`,
`EAMObjectRelationManagedBean.createNewStatus`,
`EAMObjectRelationManagedBean.editStatusSelected`,
`EAMObjectRelationManagedBean.factory`,
`EAMObjectRelationManagedBean.newStatusSelected`,
`EAMObjectRelationManagedBean.statusCount`,
`EAMObjectRelationManagedBean.statusCounter`, `EAMStaticClass.statusDAOInterface`
und `EAMObjectRelationManagedBean.tempStatus`.

String editTime ()

Holt die Informationen zu der Zeit und überprüft ob es ein Zeitpunkt oder ein -raum ist.

Rückgabe:

Definiert in Zeile 1385 der Datei `EAMObjectRelationManagedBean.java`.

Benutzt `EAMObjectRelationManagedBean.factory`,
`EAMObjectRelationManagedBean.newPeriodTimeSelected`,
`EAMObjectRelationManagedBean.newPointTimeSelected`,
`EAMObjectRelationManagedBean.showCurrentTime`,
`EAMObjectRelationManagedBean.showCurrentTimeButton`,
`EAMObjectRelationManagedBean.showPeriodTime`,
`EAMObjectRelationManagedBean.showPointTime`,
`EAMObjectRelationManagedBean.tempObjectRelation`,
`EAMObjectRelationManagedBean.tempPeriodTime`,
`EAMObjectRelationManagedBean.tempPointTime` und
`EAMStaticClass.timeDAOInterface`.

Wird benutzt von `EAMObjectRelationManagedBean.isEditTime()`.

String editType ()

Holt die Informationen des zu editierenden Typs aus der DAO und stellt diese bereit.

Rückgabe:

Definiert in Zeile 1328 der Datei `EAMObjectRelationManagedBean.java`.

Benutzt `EAMObjectRelationManagedBean.checkDeleteSelected`,

EAMObjectRelationManagedBean.createNewType,
EAMObjectRelationManagedBean.editTypeSelected,
EAMObjectRelationManagedBean.factory,
EAMObjectRelationManagedBean.newTypeSelected,
EAMStaticClass.objectTypeDAOInterface,
EAMObjectRelationManagedBean.possibleTypes,
EAMStaticClass.relationTypeDAOInterface,
EAMObjectRelationManagedBean.tempType,
EAMObjectRelationManagedBean.typeCount und
EAMObjectRelationManagedBean.typeCounter.

Calendar getCal ()

Definiert in Zeile 2247 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.cal.

Calendar getCal2 ()

Definiert in Zeile 2239 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.cal2.

long getCheckDelete ()

Definiert in Zeile 1789 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDelete.

EAMSuperClass getCurrentObjectRelation ()

Definiert in Zeile 1870 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.currentObjectRelation.

ArrayList<EAMObject> getCurrentObjects ()

Holt die vorhandenen Objekte des ausgewählten Metamodells aus der DAO.

Rückgabe:

Arraylist mit den Objekten

Definiert in Zeile 563 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMStaticClass.currentCategory,
EAMObjectRelationManagedBean.currentObjects,
EAMObjectRelationManagedBean.factory, EAMStaticClass.newRequest und
EAMStaticClass.objectDAOInterface.

Wird benutzt von EAMObjectRelationManagedBean.backToObjects().

ArrayList<EAMRelation> getCurrentRelations ()

Holt die vorhandenen Relationen des ausgewählten Metamodells aus der DAO.

Rückgabe:

Arraylist mit den Relationen

Definiert in Zeile 589 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.createNewRelation, EAMStaticClass.currentCategory, EAMObjectRelationManagedBean.currentRelations, EAMObjectRelationManagedBean.factory, EAMStaticClass.newRequest und EAMStaticClass.relationDAOInterface.

String getDay ()

Definiert in Zeile 2177 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.day.

String getDay2 ()

Definiert in Zeile 2251 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.day2.

String getErrorMsg ()

Definiert in Zeile 1821 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.errorMsg.

javax.faces.component.html.HtmlInputHidden getFirstRelationMemberFromOtherMetamodelsCounter ()

Definiert in Zeile 1934 der Datei EAMObjectRelationManagedBean.java.

Benutzt

EAMObjectRelationManagedBean.firstRelationMemberFromOtherMetamodelsCounter.

javax.faces.component.html.HtmlInputHidden getFirstRelationMemberFromOwnMetamodelCounter ()

Definiert in Zeile 1925 der Datei EAMObjectRelationManagedBean.java.

Benutzt

EAMObjectRelationManagedBean.firstRelationMemberFromOwnMetamodelCounter.

String getFirstRelationMemberMultiplicity ()

Definiert in Zeile 2034 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.firstRelationMemberMultiplicity.

Wird benutzt von EAMObjectRelationManagedBean.saveObjectRelation().

String getMonth ()

Definiert in Zeile 2189 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.month.

String getMonth2 ()

Definiert in Zeile 2259 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.month2.

EAMStatus getNewStatus ()

Definiert in Zeile 1809 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.tempStatus.

javax.faces.component.html.HtmlInputHidden getObjectCounter ()

Definiert in Zeile 1878 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.objectCounter.

ArrayList<EAMObject> getPossibleRelationsFromOtherMetamodels ()

Holt alle anderen möglichen Relationsobjekte aus den anderen Metamodellen aus der DAO und stellt diese bereit.

Rückgabe:

Definiert in Zeile 992 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.factory, EAMStaticClass.objectDAOInterface, EAMObjectRelationManagedBean.possibleRelationsFromOtherMetamodels und EAMObjectRelationManagedBean.possibleRelationsFromOwnMetamodel.

ArrayList<EAMObject> getPossibleRelationsFromOwnMetamodel ()

Holt alle möglichen Relationsobjekte aus dem eigenen Metamodell und stellt diese bereit.

Definiert in Zeile 970 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMStaticClass.currentCategory, EAMObjectRelationManagedBean.factory, EAMStaticClass.objectDAOInterface und EAMObjectRelationManagedBean.possibleRelationsFromOwnMetamodel.

ArrayList<EAMStatus> getPossibleStatuses () throws InstantiationException, IllegalAccessException

Holt alle Statusse aus der DAO und zeigt diese je nach vorhandenem Recht an.

Rückgabe:

Ausnahmebehandlung:

InstantiationException

IllegalAccessException

Definiert in Zeile 924 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.checkDeleteSelected,
EAMObjectRelationManagedBean.factory, EAMStaticClass.newRequest,
EAMObjectRelationManagedBean.noDelete(),
EAMObjectRelationManagedBean.possibleStatuses,
EAMObjectRelationManagedBean.statusCount und EAMStaticClass.statusDAOInterface.

ArrayList<EAMType> getPossibleTypes ()

Holt alle möglichen Objekt-/Relationstypen aus der DAO und stellt diese bereit.

Definiert in Zeile 1113 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.factory, EAMStaticClass.newRequest,
EAMStaticClass.objectTypeDAOInterface,
EAMObjectRelationManagedBean.possibleTypes,
EAMStaticClass.relationTypeDAOInterface,
EAMObjectRelationManagedBean.tempObjectRelation und
EAMObjectRelationManagedBean.typeCount.

javax.faces.component.html.HtmlInputHidden getRelationCounter ()

Definiert in Zeile 1887 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.relationCounter.

javax.faces.component.html.HtmlInputHidden getRelationMemberCounter ()

Definiert in Zeile 1896 der Datei EAMObjectRelationManagedBean.java.

Benutzt

EAMObjectRelationManagedBean.firstRelationMemberFromOwnMetamodelCounter.

javax.faces.component.html.HtmlInputHidden getRowCounter ()

Definiert in Zeile 1797 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.objectCounter.

**javax.faces.component.html.HtmlInputHidden
getSecondRelationMemberFromOtherMetamodelsCounter ()**

Definiert in Zeile 1952 der Datei EAMObjectRelationManagedBean.java.

Benutzt

EAMObjectRelationManagedBean.secondRelationMemberFromOtherMetamodelsCounter.

**javax.faces.component.html.HtmlInputHidden
getSecondRelationMemberFromOwnMetamodelCounter ()**

Definiert in Zeile 1943 der Datei EAMObjectRelationManagedBean.java.

Benutzt

EAMObjectRelationManagedBean.secondRelationMemberFromOwnMetamodelCounter.

String getSecondRelationMemberMultiplicity ()

Definiert in Zeile 2043 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.secondRelationMemberMultiplicity.

Wird benutzt von EAMObjectRelationManagedBean.saveObjectRelation().

String getSortColumnOver ()

Definiert in Zeile 2052 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.sortColumnOver.

int getStatusCount ()

Definiert in Zeile 2151 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.statusCount.

javax.faces.component.html.HtmlInputHidden getStatusCounter ()

Definiert in Zeile 1977 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.statusCounter.

EAMSuperClass getTempObjectRelation ()

Definiert in Zeile 1986 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.tempObjectRelation.

EAMPeriodTime getTempPeriodTime ()

Definiert in Zeile 2135 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.tempPeriodTime.

EAMPointTime getTempPointTime ()

Definiert in Zeile 2139 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.tempPointTime.

EAMStatus getTempStatus ()

Definiert in Zeile 2002 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.tempStatus.

EAMTime getTempTime ()

Definiert in Zeile 2291 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.tempTime.

EAMType getTempType ()

Definiert in Zeile 1994 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.tempType.

int getTypeCount ()

Definiert in Zeile 2159 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.typeCount.

javax.faces.component.html.HtmlInputHidden getTypeCounter ()

Definiert in Zeile 1905 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.typeCounter.

String getYear ()

Definiert in Zeile 2201 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.year.

String getYear2 ()

Definiert in Zeile 2267 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.year2.

boolean isCheckDeleteSelected ()

Definiert in Zeile 2026 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteSelected.

boolean isCheckDeleteStatus ()

Definiert in Zeile 2143 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteStatus().

boolean isCheckDeleteTime ()

Definiert in Zeile 2283 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteTime().

boolean isCheckDeleteType ()

Definiert in Zeile 2167 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteType().

boolean isCreateNewObject ()

Definiert in Zeile 2068 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.createNewObject.

boolean isCreateNewRelation ()

Definiert in Zeile 2076 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.createNewRelation.

boolean isCreateNewStatus ()

Definiert in Zeile 2092 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.createNewStatus.

boolean isCreateNewTime ()

Definiert in Zeile 2100 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.createNewTime.

boolean isCreateNewType ()

Definiert in Zeile 2084 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.createNewType.

boolean isEditObjectTypeSelected ()

Definiert in Zeile 1845 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.editTypeSelected.

boolean isEditStatusSelected ()

Definiert in Zeile 1857 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.editStatusSelected.

boolean isEditTime ()

Definiert in Zeile 2275 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.editTime().

boolean isEditTypeSelected ()

Definiert in Zeile 1969 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.editTypeSelected.

boolean isFirstObjectOtherSelected ()

Definiert in Zeile 2307 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.firstObjectOtherSelected.

boolean isNewObjectTypeSelected ()

Definiert in Zeile 1837 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.newTypeSelected.

boolean isNewPeriodTimeSelected ()

Definiert in Zeile 2116 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.newPeriodTimeSelected.

boolean isNewPointTimeSelected ()

Definiert in Zeile 2108 der Datei EAMObjectRelationManagedBean.java.

Benutzt `EAMObjectRelationManagedBean.newPointTimeSelected`.

boolean isNewStatusSelected ()

Definiert in Zeile 1829 der Datei `EAMObjectRelationManagedBean.java`.

Benutzt `EAMObjectRelationManagedBean.newStatusSelected`.

boolean isNewTypeSelected ()

Definiert in Zeile 1961 der Datei `EAMObjectRelationManagedBean.java`.

Benutzt `EAMObjectRelationManagedBean.newTypeSelected`.

boolean isObjectType ()

Definiert in Zeile 2018 der Datei `EAMObjectRelationManagedBean.java`.

Benutzt `EAMObjectRelationManagedBean.objectType`.

boolean isPeriodTime ()

Definiert in Zeile 1781 der Datei `EAMObjectRelationManagedBean.java`.

Benutzt `EAMObjectRelationManagedBean.periodTime`.

boolean isRelationType ()

Definiert in Zeile 2010 der Datei `EAMObjectRelationManagedBean.java`.

Benutzt `EAMObjectRelationManagedBean.relationType`.

boolean isShowCurrentTime ()

Definiert in Zeile 2215 der Datei `EAMObjectRelationManagedBean.java`.

Benutzt `EAMObjectRelationManagedBean.showCurrentTime`.

boolean isShowCurrentTimeButton ()

Definiert in Zeile 2299 der Datei `EAMObjectRelationManagedBean.java`.

Benutzt `EAMObjectRelationManagedBean.showCurrentTimeButton`.

boolean isShowPeriodTime ()

Definiert in Zeile 2231 der Datei `EAMObjectRelationManagedBean.java`.

Benutzt `EAMObjectRelationManagedBean.showPeriodTime`.

boolean isShowPointTime ()

Definiert in Zeile 2223 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.showPointTime.

boolean isSortAscendingOver ()

Definiert in Zeile 2060 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.sortAscendingOver.

String newObject ()

Definiert in Zeile 110 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteSelected,
EAMObjectRelationManagedBean.createNewTime, EAMStaticClass.currentCategory,
EAMStaticClass.currentObjectRelation, EAMObjectRelationManagedBean.day,
EAMObjectRelationManagedBean.day2,
EAMObjectRelationManagedBean.editStatusSelected,
EAMObjectRelationManagedBean.editTypeSelected,
EAMObjectRelationManagedBean.factory, EAMObjectRelationManagedBean.month,
EAMObjectRelationManagedBean.month2,
EAMObjectRelationManagedBean.newPeriodTimeSelected,
EAMObjectRelationManagedBean.newPointTimeSelected,
EAMObjectRelationManagedBean.newStatusSelected,
EAMObjectRelationManagedBean.newTypeSelected,
EAMObjectRelationManagedBean.objectType,
EAMObjectRelationManagedBean.relationType,
EAMObjectRelationManagedBean.showCurrentTime,
EAMObjectRelationManagedBean.showCurrentTimeButton,
EAMObjectRelationManagedBean.showPeriodTime,
EAMObjectRelationManagedBean.showPointTime,
EAMObjectRelationManagedBean.tempObjectRelation,
EAMObjectRelationManagedBean.year und EAMObjectRelationManagedBean.year2.

String newPeriodTime ()

Definiert in Zeile 1369 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.createNewTime,
EAMObjectRelationManagedBean.newPeriodTimeSelected,
EAMObjectRelationManagedBean.showCurrentTime,
EAMObjectRelationManagedBean.showCurrentTimeButton,
EAMObjectRelationManagedBean.showPeriodTime und
EAMObjectRelationManagedBean.tempPeriodTime.

String newPointTime ()

Definiert in Zeile 1357 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.createNewTime,
EAMObjectRelationManagedBean.newPointTimeSelected,
EAMObjectRelationManagedBean.showCurrentTime,
EAMObjectRelationManagedBean.showCurrentTimeButton,
EAMObjectRelationManagedBean.showPointTime und
EAMObjectRelationManagedBean.tempPointTime.

String newRelation ()

Definiert in Zeile 142 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteSelected,
EAMObjectRelationManagedBean.createNewRelation,
EAMObjectRelationManagedBean.createNewTime, EAMStaticClass.currentCategory,
EAMStaticClass.currentObjectRelation, EAMObjectRelationManagedBean.day,
EAMObjectRelationManagedBean.day2,
EAMObjectRelationManagedBean.editStatusSelected,
EAMObjectRelationManagedBean.editTypeSelected,
EAMObjectRelationManagedBean.factory,
EAMObjectRelationManagedBean.firstRelationMemberMultiplicity,
EAMObjectRelationManagedBean.month, EAMObjectRelationManagedBean.month2,
EAMObjectRelationManagedBean.newPeriodTimeSelected,
EAMObjectRelationManagedBean.newPointTimeSelected,
EAMObjectRelationManagedBean.newStatusSelected,
EAMObjectRelationManagedBean.newTypeSelected,
EAMObjectRelationManagedBean.objectType,
EAMObjectRelationManagedBean.relationType,
EAMObjectRelationManagedBean.secondRelationMemberMultiplicity,
EAMObjectRelationManagedBean.showCurrentTime,
EAMObjectRelationManagedBean.showCurrentTimeButton,
EAMObjectRelationManagedBean.showPeriodTime,
EAMObjectRelationManagedBean.showPointTime,
EAMObjectRelationManagedBean.tempObjectRelation,
EAMObjectRelationManagedBean.year und EAMObjectRelationManagedBean.year2.

String newStatus ()

Definiert in Zeile 810 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteSelected,
EAMObjectRelationManagedBean.createNewStatus,
EAMObjectRelationManagedBean.editStatusSelected,
EAMObjectRelationManagedBean.newStatusSelected,
EAMObjectRelationManagedBean.statusCount und

EAMObjectRelationManagedBean.tempStatus.

String newType ()

Definiert in Zeile 1181 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteSelected,
EAMObjectRelationManagedBean.createNewType, EAMStaticClass.currentCategory,
EAMObjectRelationManagedBean.editTypeSelected,
EAMObjectRelationManagedBean.factory,
EAMObjectRelationManagedBean.newTypeSelected,
EAMObjectRelationManagedBean.tempObjectRelation,
EAMObjectRelationManagedBean.tempType und
EAMObjectRelationManagedBean.typeCount.

String noDelete ()

Definiert in Zeile 771 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteSelected,
EAMObjectRelationManagedBean.checkDeleteStatus(),
EAMObjectRelationManagedBean.checkDeleteTime(),
EAMObjectRelationManagedBean.checkDeleteType(),
EAMObjectRelationManagedBean.createNewObject,
EAMObjectRelationManagedBean.createNewRelation,
EAMObjectRelationManagedBean.createNewStatus,
EAMObjectRelationManagedBean.createNewType und
EAMObjectRelationManagedBean.showCurrentTimeButton.

Wird benutzt von EAMObjectRelationManagedBean.getPossibleStatuses().

String saveObjectRelation () throws Exception

Überprüft vor dem Speichern, ob das Objekt eine Instanz vom EAMObject oder EAMRelation ist und die Korrektheit der Eingaben. Anschliessend wird kontrolliert, ob das Objekt bereits eine ID besitzt. Falls ja, dann wird das editierte Objekt gespeichert, andernfalls bekommt das Objekt eine ID und wird in die Datenbank gespeichert.

Rückgabe:

Ausnahmebehandlung:

Exception

Definiert in Zeile 183 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.createNewRelation,
EAMStaticClass.currentCategory, EAMStaticClass.currentObjectRelation,
EAMObjectRelationManagedBean.factory,

EAMObjectRelationManagedBean.firstRelationMemberMultiplicity,
EAMObjectRelationManagedBean.getFirstRelationMemberMultiplicity(),
EAMObjectRelationManagedBean.getSecondRelationMemberMultiplicity(),
EAMStaticClass.objectDAOInterface, EAMStaticClass.relationDAOInterface,
EAMObjectRelationManagedBean.secondRelationMemberMultiplicity und
EAMObjectRelationManagedBean.tempObjectRelation.

String saveStatus () throws Exception

Speichert entweder einen editieren Status (ID > 0) oder ein neues Status in die DAO.

Rückgabe:

Ausnahmebehandlung:

Exception

Definiert in Zeile 827 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.createNewStatus,
EAMObjectRelationManagedBean.editStatusSelected,
EAMObjectRelationManagedBean.factory,
EAMObjectRelationManagedBean.newStatusSelected,
EAMObjectRelationManagedBean.possibleStatuses, EAMStaticClass.statusDAOInterface
und EAMObjectRelationManagedBean.tempStatus.

String saveTime ()

Überprüft die Korrektheit der Eingaben und speichert die Zeit, Zeitpunkt oder - raum, in die DAO

Rückgabe:

Definiert in Zeile 1431 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.cal, EAMObjectRelationManagedBean.cal2,
EAMObjectRelationManagedBean.day, EAMObjectRelationManagedBean.day2,
EAMObjectRelationManagedBean.factory, EAMObjectRelationManagedBean.month,
EAMObjectRelationManagedBean.month2,
EAMObjectRelationManagedBean.newPeriodTimeSelected,
EAMObjectRelationManagedBean.newPointTimeSelected,
EAMObjectRelationManagedBean.showCurrentTime,
EAMObjectRelationManagedBean.showCurrentTimeButton,
EAMObjectRelationManagedBean.showPeriodTime,
EAMObjectRelationManagedBean.showPointTime,
EAMObjectRelationManagedBean.tempObjectRelation,
EAMObjectRelationManagedBean.tempPeriodTime,

EAMObjectRelationManagedBean.tempPointTime, EAMStaticClass.timeDAOInterface, EAMObjectRelationManagedBean.year und EAMObjectRelationManagedBean.year2.

String saveType () throws Exception

Prüft ob die Eingaben korrekt sind und ob der Typ editiert wurde. Anschließend wird der Typ in die DAO gespeichert.

Rückgabe:

Ausnahmebehandlung:

Exception

Definiert in Zeile 1209 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.checkDeleteSelected, EAMObjectRelationManagedBean.createNewType, EAMObjectRelationManagedBean.editTypeSelected, EAMObjectRelationManagedBean.factory, EAMObjectRelationManagedBean.newTypeSelected, EAMStaticClass.objectTypeDAOInterface, EAMObjectRelationManagedBean.possibleTypes, EAMStaticClass.relationTypeDAOInterface und EAMObjectRelationManagedBean.tempType.

void setCal (Calendar cal)

Definiert in Zeile 2211 der Datei EAMObjectRelationManagedBean.java.

void setCal2 (Calendar cal2)

Definiert in Zeile 2243 der Datei EAMObjectRelationManagedBean.java.

void setCheckDelete (long checkDelete)

Definiert in Zeile 1793 der Datei EAMObjectRelationManagedBean.java.

void setCheckDeleteSelected (boolean checkDeleteSelected)

Definiert in Zeile 2030 der Datei EAMObjectRelationManagedBean.java.

void setCheckDeleteStatus (boolean checkDeleteStatus)

Definiert in Zeile 2147 der Datei EAMObjectRelationManagedBean.java.

void setCheckDeleteTime (boolean checkDeleteTime)

Definiert in Zeile 2287 der Datei EAMObjectRelationManagedBean.java.

void setCheckDeleteType (boolean checkDeleteType)

Definiert in Zeile 2171 der Datei EAMObjectRelationManagedBean.java.

void setCreateNewObject (boolean createNewObject)

Definiert in Zeile 2072 der Datei EAMObjectRelationManagedBean.java.

void setCreateNewRelation (boolean createNewRelation)

Definiert in Zeile 2080 der Datei EAMObjectRelationManagedBean.java.

void setCreateNewStatus (boolean createNewStatus)

Definiert in Zeile 2096 der Datei EAMObjectRelationManagedBean.java.

void setCreateNewTime (boolean createNewTime)

Definiert in Zeile 2104 der Datei EAMObjectRelationManagedBean.java.

void setCreateNewType (boolean createNewType)

Definiert in Zeile 2088 der Datei EAMObjectRelationManagedBean.java.

void setCurrentObjectRelation (EAMSuperClass currentObjectRelation)

Definiert in Zeile 1874 der Datei EAMObjectRelationManagedBean.java.

void setCurrentObjects (ArrayList< EAMObject > currentObjects)

Definiert in Zeile 1817 der Datei EAMObjectRelationManagedBean.java.

void setCurrentRelations (ArrayList< EAMRelation > currentRelations)

Definiert in Zeile 1853 der Datei EAMObjectRelationManagedBean.java.

void setDay (String day)

Definiert in Zeile 2183 der Datei EAMObjectRelationManagedBean.java.

void setDay2 (String day2)

Definiert in Zeile 2255 der Datei EAMObjectRelationManagedBean.java.

void setEditObjectTypeSelected (boolean editObjectTypeSelected)

Definiert in Zeile 1849 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.editTypeSelected.

void setEditStatusSelected (boolean editStatusSelected)

Definiert in Zeile 1861 der Datei EAMObjectRelationManagedBean.java.

void setEditTime (boolean editTime)

Definiert in Zeile 2279 der Datei EAMObjectRelationManagedBean.java.

void setEditTypeSelected (boolean editTypeSelected)

Definiert in Zeile 1973 der Datei EAMObjectRelationManagedBean.java.

void setErrorMsg (String errorMsg)

Definiert in Zeile 1825 der Datei EAMObjectRelationManagedBean.java.

void setFirstObjectOtherSelected (boolean firstObjectOtherSelected)

Definiert in Zeile 2311 der Datei EAMObjectRelationManagedBean.java.

**void setFirstRelationMemberFromOtherMetamodelsCounter
(javax.faces.component.html.HtmlInputHidden
firstRelationMemberFromOtherMetamodelsCounter)**

Definiert in Zeile 1938 der Datei EAMObjectRelationManagedBean.java.

Benutzt

EAMObjectRelationManagedBean.firstRelationMemberFromOtherMetamodelsCounter.

**void setFirstRelationMemberFromOwnMetamodelCounter
(javax.faces.component.html.HtmlInputHidden
firstRelationMemberFromOwnMetamodelCounter)**

Definiert in Zeile 1929 der Datei EAMObjectRelationManagedBean.java.

Benutzt

EAMObjectRelationManagedBean.firstRelationMemberFromOwnMetamodelCounter.

void setFirstRelationMemberMultiplicity (String firstRelationMemberMultiplicity)

Definiert in Zeile 2038 der Datei EAMObjectRelationManagedBean.java.

void setMonth (String month)

Definiert in Zeile 2195 der Datei EAMObjectRelationManagedBean.java.

void setMonth2 (String month2)

Definiert in Zeile 2263 der Datei EAMObjectRelationManagedBean.java.

void setNewObjectTypeSelected (boolean newObjectTypeSelected)

Definiert in Zeile 1841 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.newTypeSelected.

void setNewPeriodTimeSelected (boolean newPeriodTimeSelected)

Definiert in Zeile 2120 der Datei EAMObjectRelationManagedBean.java.

void setNewPointTimeSelected (boolean newPointTimeSelected)

Definiert in Zeile 2112 der Datei EAMObjectRelationManagedBean.java.

void setNewStatus (EAMStatus newStatus)

Definiert in Zeile 1813 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.tempStatus.

void setNewStatusSelected (boolean newStatusSelected)

Definiert in Zeile 1833 der Datei EAMObjectRelationManagedBean.java.

void setNewTypeSelected (boolean newTypeSelected)

Definiert in Zeile 1965 der Datei EAMObjectRelationManagedBean.java.

void setObjectCounter (javax.faces.component.html.HtmlInputHidden objectCounter)

Definiert in Zeile 1882 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.objectCounter.

void setObjectType (boolean objectType)

Definiert in Zeile 2022 der Datei EAMObjectRelationManagedBean.java.

void setPeriodTime (boolean periodTime)

Definiert in Zeile 1785 der Datei EAMObjectRelationManagedBean.java.

void setPossibleRelationsFromOtherMetamodels (ArrayList< EAMObject > possibleRelationsFromOtherMetamodels)

Definiert in Zeile 1920 der Datei EAMObjectRelationManagedBean.java.

void setPossibleRelationsFromOwnMetamodel (ArrayList< EAMObject > possibleRelationsFromOwnMetamodel)

Definiert in Zeile 1914 der Datei EAMObjectRelationManagedBean.java.

void setPossibleStatuses (ArrayList< EAMStatus > possibleStatuses)

Definiert in Zeile 1805 der Datei EAMObjectRelationManagedBean.java.

void setPossibleTypes (ArrayList< EAMType > possibleTypes)

Definiert in Zeile 1865 der Datei EAMObjectRelationManagedBean.java.

void setRelationCounter (javax.faces.component.html.HtmlInputHidden relationCounter)

Definiert in Zeile 1891 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.relationCounter.

void setRelationMemberCounter (javax.faces.component.html.HtmlInputHidden relationMemberCounter)

Definiert in Zeile 1900 der Datei EAMObjectRelationManagedBean.java.

Benutzt

EAMObjectRelationManagedBean.firstRelationMemberFromOwnMetamodelCounter.

void setRelationType (boolean relationType)

Definiert in Zeile 2014 der Datei EAMObjectRelationManagedBean.java.

void setRowCounter (javax.faces.component.html.HtmlInputHidden rowCounter)

Definiert in Zeile 1801 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.objectCounter.

**void setSecondRelationMemberFromOtherMetamodelsCounter
(javax.faces.component.html.HtmlInputHidden
secondRelationMemberFromOtherMetamodelsCounter)**

Definiert in Zeile 1956 der Datei EAMObjectRelationManagedBean.java.

Benutzt

EAMObjectRelationManagedBean.secondRelationMemberFromOtherMetamodelsCounter.

**void setSecondRelationMemberFromOwnMetamodelCounter
(javax.faces.component.html.HtmlInputHidden
secondRelationMemberFromOwnMetamodelCounter)**

Definiert in Zeile 1947 der Datei EAMObjectRelationManagedBean.java.

Benutzt

EAMObjectRelationManagedBean.secondRelationMemberFromOwnMetamodelCounter.

void setSecondRelationMemberMultiplicity (String secondRelationMemberMultiplicity)

Definiert in Zeile 2047 der Datei EAMObjectRelationManagedBean.java.

void setShowCurrentTime (boolean showCurrentTime)

Definiert in Zeile 2219 der Datei EAMObjectRelationManagedBean.java.

void setShowCurrentTimeButton (boolean showCurrentTimeButton)

Definiert in Zeile 2303 der Datei EAMObjectRelationManagedBean.java.

void setShowPeriodTime (boolean showPeriodTime)

Definiert in Zeile 2235 der Datei EAMObjectRelationManagedBean.java.

void setShowPointTime (boolean showPointTime)

Definiert in Zeile 2227 der Datei EAMObjectRelationManagedBean.java.

void setSortAscendingOver (boolean sortAscendingOver)

Definiert in Zeile 2064 der Datei EAMObjectRelationManagedBean.java.

void setSortColumnOver (String sortColumnOver)

Definiert in Zeile 2056 der Datei EAMObjectRelationManagedBean.java.

void setStatusCount (int statusCount)

Definiert in Zeile 2155 der Datei EAMObjectRelationManagedBean.java.

void setStatusCounter (javax.faces.component.html.HtmlInputHidden statusCounter)

Definiert in Zeile 1981 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.statusCounter.

void setTempObjectRelation (EAMSuperClass tempObjectRelation)

Definiert in Zeile 1990 der Datei EAMObjectRelationManagedBean.java.

void setTempPeriodTime (EAMPeriodTime tempPeriodTime)

Definiert in Zeile 2125 der Datei EAMObjectRelationManagedBean.java.

void setTempPointTime (EAMPointTime tempPointTime)

Definiert in Zeile 2131 der Datei EAMObjectRelationManagedBean.java.

void setTempStatus (EAMStatus tempStatus)

Definiert in Zeile 2006 der Datei EAMObjectRelationManagedBean.java.

void setTempTime (EAMTime tempTime)

Definiert in Zeile 2295 der Datei EAMObjectRelationManagedBean.java.

void setTempType (EAMType tempType)

Definiert in Zeile 1998 der Datei EAMObjectRelationManagedBean.java.

void setTypeCount (int typeCount)

Definiert in Zeile 2163 der Datei EAMObjectRelationManagedBean.java.

void setTypeCounter (javax.faces.component.html.HtmlInputHidden typeCounter)

Definiert in Zeile 1909 der Datei EAMObjectRelationManagedBean.java.

Benutzt EAMObjectRelationManagedBean.typeCounter.

void setYear (String year)

Definiert in Zeile 2207 der Datei EAMObjectRelationManagedBean.java.

void setYear2 (String year2)

Definiert in Zeile 2271 der Datei EAMObjectRelationManagedBean.java.

4.4.3.3 Dokumentation der Datenelemente**Calendar cal [private]**

Definiert in Zeile 40 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.getCal() und EAMObjectRelationManagedBean.saveTime().

Calendar cal2 [private]

Definiert in Zeile 41 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.getCal2() und EAMObjectRelationManagedBean.saveTime().

long checkDelete [private]

Definiert in Zeile 65 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.checkDeleteStatus(), EAMObjectRelationManagedBean.checkDeleteTime(), EAMObjectRelationManagedBean.deleteStatus(), EAMObjectRelationManagedBean.deleteTime() und EAMObjectRelationManagedBean.getCheckDelete().

boolean checkDeleteSelected [private]

Definiert in Zeile 74 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.backToObject(), EAMObjectRelationManagedBean.checkDeleteObject(), EAMObjectRelationManagedBean.checkDeleteRelation(), EAMObjectRelationManagedBean.checkDeleteStatus(), EAMObjectRelationManagedBean.deleteObjectRelation(), EAMObjectRelationManagedBean.deleteStatus(), EAMObjectRelationManagedBean.editObject(), EAMObjectRelationManagedBean.editRelation(), EAMObjectRelationManagedBean.editStatus(), EAMObjectRelationManagedBean.editType(), EAMObjectRelationManagedBean.getPossibleStatuses(), EAMObjectRelationManagedBean.isCheckDeleteSelected(), EAMObjectRelationManagedBean.newObject(), EAMObjectRelationManagedBean.newRelation(), EAMObjectRelationManagedBean.newStatus(),

EAMObjectRelationManagedBean.newType(),
EAMObjectRelationManagedBean.noDelete() und
EAMObjectRelationManagedBean.saveType().

boolean checkDeleteStatus = false [private]

Definiert in Zeile 68 der Datei EAMObjectRelationManagedBean.java.

boolean checkDeleteTime = false [private]

Definiert in Zeile 87 der Datei EAMObjectRelationManagedBean.java.

boolean checkDeleteType = false [private]

Definiert in Zeile 69 der Datei EAMObjectRelationManagedBean.java.

boolean createNewObject = true [private]

Definiert in Zeile 75 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.backToObjects(),
EAMObjectRelationManagedBean.checkDeleteObject(),
EAMObjectRelationManagedBean.deleteObjectRelation(),
EAMObjectRelationManagedBean.isCreateNewObject() und
EAMObjectRelationManagedBean.noDelete().

boolean createNewRelation = true [private]

Definiert in Zeile 76 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.backToObjects(),
EAMObjectRelationManagedBean.checkDeleteRelation(),
EAMObjectRelationManagedBean.deleteObjectRelation(),
EAMObjectRelationManagedBean.getCurrentRelations(),
EAMObjectRelationManagedBean.isCreateNewRelation(),
EAMObjectRelationManagedBean.newRelation(),
EAMObjectRelationManagedBean.noDelete() und
EAMObjectRelationManagedBean.saveObjectRelation().

boolean createNewStatus = true [private]

Definiert in Zeile 78 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.checkDeleteStatus(),
EAMObjectRelationManagedBean.deleteStatus(),

EAMObjectRelationManagedBean.editStatus(),
EAMObjectRelationManagedBean.isCreateNewStatus(),
EAMObjectRelationManagedBean.newStatus(),
EAMObjectRelationManagedBean.noDelete() und
EAMObjectRelationManagedBean.saveStatus().

boolean createNewTime = true [private]

Definiert in Zeile 79 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.checkDeleteTime(),
EAMObjectRelationManagedBean.deleteTime(),
EAMObjectRelationManagedBean.editObject(),
EAMObjectRelationManagedBean.editRelation(),
EAMObjectRelationManagedBean.isCreateNewTime(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newPeriodTime(),
EAMObjectRelationManagedBean.newPointTime() und
EAMObjectRelationManagedBean.newRelation().

boolean createNewType = true [private]

Definiert in Zeile 77 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.checkDeleteType(),
EAMObjectRelationManagedBean.deleteType(),
EAMObjectRelationManagedBean.editType(),
EAMObjectRelationManagedBean.isCreateNewType(),
EAMObjectRelationManagedBean.newType(),
EAMObjectRelationManagedBean.noDelete() und
EAMObjectRelationManagedBean.saveType().

EAMSuperClass currentObjectRelation [private]

Definiert in Zeile 45 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.checkDeleteObject(),
EAMObjectRelationManagedBean.checkDeleteRelation(),
EAMObjectRelationManagedBean.deleteObjectRelation() und
EAMObjectRelationManagedBean.getCurrentObjectRelation().

ArrayList<EAMObject> currentObjects = new ArrayList<EAMObject>()
[private]

Definiert in Zeile 54 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von `EAMObjectRelationManagedBean.backToObjects()` und `EAMObjectRelationManagedBean.getCurrentObjects()`.

ArrayList<EAMRelation> currentRelations = new ArrayList<EAMRelation>()
[private]

Definiert in Zeile 97 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.getCurrentRelations()`.

EAMType currentType [private]

Definiert in Zeile 98 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.checkDeleteType()` und `EAMObjectRelationManagedBean.deleteType()`.

String day [private]

Definiert in Zeile 100 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.getDay()`,
`EAMObjectRelationManagedBean.newObject()`,
`EAMObjectRelationManagedBean.newRelation()` und
`EAMObjectRelationManagedBean.saveTime()`.

String day2 [private]

Definiert in Zeile 101 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.getDay2()`,
`EAMObjectRelationManagedBean.newObject()`,
`EAMObjectRelationManagedBean.newRelation()` und
`EAMObjectRelationManagedBean.saveTime()`.

boolean editStatusSelected = false [private]

Definiert in Zeile 67 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.backToObject()`,
`EAMObjectRelationManagedBean.checkDeleteStatus()`,
`EAMObjectRelationManagedBean.editObject()`,
`EAMObjectRelationManagedBean.editRelation()`,
`EAMObjectRelationManagedBean.editStatus()`,
`EAMObjectRelationManagedBean.isEditStatusSelected()`,
`EAMObjectRelationManagedBean.newObject()`,
`EAMObjectRelationManagedBean.newRelation()`,
`EAMObjectRelationManagedBean.newStatus()` und
`EAMObjectRelationManagedBean.saveStatus()`.

boolean editTime = false [private]

Definiert in Zeile 86 der Datei EAMObjectRelationManagedBean.java.

boolean editTypeSelected = false [private]

Definiert in Zeile 71 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.checkDeleteType(),
EAMObjectRelationManagedBean.editObject(),
EAMObjectRelationManagedBean.editRelation(),
EAMObjectRelationManagedBean.editType(),
EAMObjectRelationManagedBean.isEditObjectTypeSelected(),
EAMObjectRelationManagedBean.isEditTypeSelected(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newRelation(),
EAMObjectRelationManagedBean.newType(),
EAMObjectRelationManagedBean.saveType() und
EAMObjectRelationManagedBean.setEditObjectTypeSelected().

String errorMsg [private]

Definiert in Zeile 44 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.getErrorMsg().

EAMStaticClass factory = EAMStaticClass.getInstance(userId, password, this.getClass().getName()) [private]

Definiert in Zeile 39 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.checkDeleteObject(),
EAMObjectRelationManagedBean.checkDeleteRelation(),
EAMObjectRelationManagedBean.checkDeleteStatus(),
EAMObjectRelationManagedBean.checkDeleteTime(),
EAMObjectRelationManagedBean.checkDeleteType(),
EAMObjectRelationManagedBean.deleteObjectRelation(),
EAMObjectRelationManagedBean.deleteStatus(),
EAMObjectRelationManagedBean.deleteTime(),
EAMObjectRelationManagedBean.deleteType(),
EAMObjectRelationManagedBean.editObject(),
EAMObjectRelationManagedBean.editRelation(),
EAMObjectRelationManagedBean.editStatus(),
EAMObjectRelationManagedBean.editTime(),
EAMObjectRelationManagedBean.editType(),
EAMObjectRelationManagedBean.getCurrentObjects(),
EAMObjectRelationManagedBean.getCurrentRelations(),

EAMObjectRelationManagedBean.getPossibleRelationsFromOtherMetamodels(),
EAMObjectRelationManagedBean.getPossibleRelationsFromOwnMetamodel(),
EAMObjectRelationManagedBean.getPossibleStatuses(),
EAMObjectRelationManagedBean.getPossibleTypes(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newRelation(),
EAMObjectRelationManagedBean.newType(),
EAMObjectRelationManagedBean.saveObjectRelation(),
EAMObjectRelationManagedBean.saveStatus(),
EAMObjectRelationManagedBean.saveTime() und
EAMObjectRelationManagedBean.saveType().

boolean firstObjectOtherSelected = true [private]

Definiert in Zeile 88 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.addFirstRelationMember() und
EAMObjectRelationManagedBean.isFirstObjectOtherSelected().

javax.faces.component.html.HtmlInputHidden

firstRelationMemberFromOtherMetamodelsCounter [private]

Definiert in Zeile 51 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.addFirstRelationMember(), EAM-
ObjectRelationManagedBean.getFirstRelationMemberFromOtherMetamodelsCounter()
und EAMObjectRelationManagedBean.setFirstRelationMemberFromOtherMetamodelsCounter().

javax.faces.component.html.HtmlInputHidden

firstRelationMemberFromOwnMetamodelCounter [private]

Definiert in Zeile 50 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.addFirstRelationMember(), EAM-
ObjectRelationManagedBean.getFirstRelationMemberFromOwnMetamodelCounter(),
EAMObjectRelationManagedBean.getRelationMemberCounter(),
EAMObjectRelationManagedBean.setFirstRelationMemberFromOwnMetamodelCounter()
und EAMObjectRelationManagedBean.setRelationMemberCounter().

String firstRelationMemberMultiplicity [private]

Definiert in Zeile 62 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.editRelation(),
EAMObjectRelationManagedBean.getFirstRelationMemberMultiplicity(),
EAMObjectRelationManagedBean.newRelation() und
EAMObjectRelationManagedBean.saveObjectRelation().

String month [private]

Definiert in Zeile 102 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.getMonth(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newRelation() und
EAMObjectRelationManagedBean.saveTime().

String month2 [private]

Definiert in Zeile 103 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.getMonth2(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newRelation() und
EAMObjectRelationManagedBean.saveTime().

boolean newPeriodTimeSelected = false [private]

Definiert in Zeile 81 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.editObject(),
EAMObjectRelationManagedBean.editRelation(),
EAMObjectRelationManagedBean.editTime(),
EAMObjectRelationManagedBean.isNewPeriodTimeSelected(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newPeriodTime(),
EAMObjectRelationManagedBean.newRelation() und
EAMObjectRelationManagedBean.saveTime().

boolean newPointTimeSelected = false [private]

Definiert in Zeile 80 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.editObject(),
EAMObjectRelationManagedBean.editRelation(),
EAMObjectRelationManagedBean.editTime(),
EAMObjectRelationManagedBean.isNewPointTimeSelected(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newPointTime(),
EAMObjectRelationManagedBean.newRelation() und
EAMObjectRelationManagedBean.saveTime().

boolean newStatusSelected = false [private]

Definiert in Zeile 66 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von `EAMObjectRelationManagedBean.backToObject()`,
`EAMObjectRelationManagedBean.checkDeleteStatus()`,
`EAMObjectRelationManagedBean.editObject()`,
`EAMObjectRelationManagedBean.editRelation()`,
`EAMObjectRelationManagedBean.editStatus()`,
`EAMObjectRelationManagedBean.isNewStatusSelected()`,
`EAMObjectRelationManagedBean.newObject()`,
`EAMObjectRelationManagedBean.newRelation()`,
`EAMObjectRelationManagedBean.newStatus()` und
`EAMObjectRelationManagedBean.saveStatus()`.

boolean newTypeSelected = false [private]

Definiert in Zeile 70 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.backToObject()`,
`EAMObjectRelationManagedBean.editObject()`,
`EAMObjectRelationManagedBean.editRelation()`,
`EAMObjectRelationManagedBean.editType()`,
`EAMObjectRelationManagedBean.isNewObjectTypeSelected()`,
`EAMObjectRelationManagedBean.isNewTypeSelected()`,
`EAMObjectRelationManagedBean.newObject()`,
`EAMObjectRelationManagedBean.newRelation()`,
`EAMObjectRelationManagedBean.newType()`,
`EAMObjectRelationManagedBean.saveType()` und
`EAMObjectRelationManagedBean.setNewObjectTypeSelected()`.

javax.faces.component.html.HtmlInputHidden objectCounter [private]

Definiert in Zeile 46 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.checkDeleteObject()`,
`EAMObjectRelationManagedBean.editObject()`,
`EAMObjectRelationManagedBean.getObjectCounter()`,
`EAMObjectRelationManagedBean.getRowCounter()`,
`EAMObjectRelationManagedBean.setObjectCounter()` und
`EAMObjectRelationManagedBean.setRowCounter()`.

boolean objectType [private]

Definiert in Zeile 73 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.checkDeleteType()`,
`EAMObjectRelationManagedBean.deleteType()`,
`EAMObjectRelationManagedBean.editObject()`,
`EAMObjectRelationManagedBean.editRelation()`,
`EAMObjectRelationManagedBean.isObjectType()`,

EAMObjectRelationManagedBean.newObject() und
EAMObjectRelationManagedBean.newRelation().

String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")

Definiert in Zeile 37 der Datei EAMObjectRelationManagedBean.java.

boolean periodTime [private]

Definiert in Zeile 64 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.isPeriodTime().

ArrayList<EAMObject> possibleRelationsFromOtherMetamodels = new
ArrayList<EAMObject>() [private]

Definiert in Zeile 96 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.addFirstRelationMember(),
EAMObjectRelationManagedBean.addSecondRelationMember() und
EAMObjectRelationManagedBean.getPossibleRelationsFromOtherMetamodels().

ArrayList<EAMObject> possibleRelationsFromOwnMetamodel = new
ArrayList<EAMObject>() [private]

Definiert in Zeile 95 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.addFirstRelationMember(),
EAMObjectRelationManagedBean.addSecondRelationMember(),
EAMObjectRelationManagedBean.getPossibleRelationsFromOtherMetamodels() und
EAMObjectRelationManagedBean.getPossibleRelationsFromOwnMetamodel().

ArrayList<EAMStatus> possibleStatuses = new ArrayList<EAMStatus>()
[private]

Definiert in Zeile 57 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.addStatus(),
EAMObjectRelationManagedBean.deleteStatus(),
EAMObjectRelationManagedBean.getPossibleStatuses() und
EAMObjectRelationManagedBean.saveStatus().

ArrayList<EAMType> possibleTypes = new ArrayList<EAMType>() [private]

Definiert in Zeile 55 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von `EAMObjectRelationManagedBean.addType()`,
`EAMObjectRelationManagedBean.deleteType()`,
`EAMObjectRelationManagedBean.editType()`,
`EAMObjectRelationManagedBean.getPossibleTypes()` und
`EAMObjectRelationManagedBean.saveType()`.

`javax.faces.component.html.HtmlInputHidden relationCounter` [private]

Definiert in Zeile 47 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.checkDeleteRelation()`,
`EAMObjectRelationManagedBean.editRelation()`,
`EAMObjectRelationManagedBean.getRelationCounter()` und
`EAMObjectRelationManagedBean.setRelationCounter()`.

`boolean relationType` [private]

Definiert in Zeile 72 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.editObject()`,
`EAMObjectRelationManagedBean.editRelation()`,
`EAMObjectRelationManagedBean.isRelationType()`,
`EAMObjectRelationManagedBean.newObject()` und
`EAMObjectRelationManagedBean.newRelation()`.

**`javax.faces.component.html.HtmlInputHidden
secondRelationMemberFromOtherMetamodelsCounter` [private]**

Definiert in Zeile 53 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.addSecondRelationMember()`, `EAMObjectRelationManagedBean.getSecondRelationMemberFromOtherMetamodelsCounter()` und `EAMObjectRelationManagedBean.setSecondRelationMemberFromOtherMetamodelsCounter()`.

**`javax.faces.component.html.HtmlInputHidden
secondRelationMemberFromOwnMetamodelCounter` [private]**

Definiert in Zeile 52 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.addSecondRelationMember()`, `EAMObjectRelationManagedBean.getSecondRelationMemberFromOwnMetamodelCounter()` und `EAMObjectRelationManagedBean.setSecondRelationMemberFromOwnMetamodelCounter()`.

`String secondRelationMemberMultiplicity` [private]

Definiert in Zeile 63 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von EAMObjectRelationManagedBean.editRelation(),
EAMObjectRelationManagedBean.getSecondRelationMemberMultiplicity(),
EAMObjectRelationManagedBean.newRelation() und
EAMObjectRelationManagedBean.saveObjectRelation().

boolean showCurrentTime = false [private]

Definiert in Zeile 82 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.deleteTime(),
EAMObjectRelationManagedBean.editObject(),
EAMObjectRelationManagedBean.editRelation(),
EAMObjectRelationManagedBean.editTime(),
EAMObjectRelationManagedBean.isShowCurrentTime(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newPeriodTime(),
EAMObjectRelationManagedBean.newPointTime(),
EAMObjectRelationManagedBean.newRelation() und
EAMObjectRelationManagedBean.saveTime().

boolean showCurrentTimeButton = false [private]

Definiert in Zeile 83 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.checkDeleteTime(),
EAMObjectRelationManagedBean.editObject(),
EAMObjectRelationManagedBean.editRelation(),
EAMObjectRelationManagedBean.editTime(),
EAMObjectRelationManagedBean.isShowCurrentTimeButton(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newPeriodTime(),
EAMObjectRelationManagedBean.newPointTime(),
EAMObjectRelationManagedBean.newRelation(),
EAMObjectRelationManagedBean.noDelete() und
EAMObjectRelationManagedBean.saveTime().

boolean showPeriodTime = false [private]

Definiert in Zeile 85 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.deleteTime(),
EAMObjectRelationManagedBean.editObject(),
EAMObjectRelationManagedBean.editRelation(),
EAMObjectRelationManagedBean.editTime(),
EAMObjectRelationManagedBean.isShowPeriodTime(),

EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newPeriodTime(),
EAMObjectRelationManagedBean.newRelation() und
EAMObjectRelationManagedBean.saveTime().

boolean showPointTime = false [private]

Definiert in Zeile 84 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.deleteTime(),
EAMObjectRelationManagedBean.editObject(),
EAMObjectRelationManagedBean.editRelation(),
EAMObjectRelationManagedBean.editTime(),
EAMObjectRelationManagedBean.isShowPointTime(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newPointTime(),
EAMObjectRelationManagedBean.newRelation() und
EAMObjectRelationManagedBean.saveTime().

boolean sortAscendingOver = true [private]

Definiert in Zeile 90 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.isSortAscendingOver().

String sortColumnOver = null [private]

Definiert in Zeile 89 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.getSortColumnOver().

int statusCount = 0 [private]

Definiert in Zeile 92 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.checkDeleteStatus(),
EAMObjectRelationManagedBean.editStatus(),
EAMObjectRelationManagedBean.getPossibleStatuses(),
EAMObjectRelationManagedBean.getStatusCount() und
EAMObjectRelationManagedBean.newStatus().

javax.faces.component.html.HtmlInputHidden statusCounter [private]

Definiert in Zeile 48 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.addStatus(),
EAMObjectRelationManagedBean.checkDeleteStatus(),
EAMObjectRelationManagedBean.editStatus(),

EAMObjectRelationManagedBean.getStatusCounter() und
EAMObjectRelationManagedBean.setStatusCounter().

EAMSuperClass tempObjectRelation [private]

Definiert in Zeile 43 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.addFirstRelationMember(),
EAMObjectRelationManagedBean.addSecondRelationMember(),
EAMObjectRelationManagedBean.addStatus(),
EAMObjectRelationManagedBean.addType(),
EAMObjectRelationManagedBean.backToObject(),
EAMObjectRelationManagedBean.checkDeleteObject(),
EAMObjectRelationManagedBean.checkDeleteRelation(),
EAMObjectRelationManagedBean.checkDeleteTime(),
EAMObjectRelationManagedBean.deleteTime(),
EAMObjectRelationManagedBean.editObject(),
EAMObjectRelationManagedBean.editRelation(),
EAMObjectRelationManagedBean.editTime(),
EAMObjectRelationManagedBean.getPossibleTypes(),
EAMObjectRelationManagedBean.getTempObjectRelation(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newRelation(),
EAMObjectRelationManagedBean.newType(),
EAMObjectRelationManagedBean.saveObjectRelation() und
EAMObjectRelationManagedBean.saveTime().

EAMPeriodTime tempPeriodTime [private]

Definiert in Zeile 59 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.editTime(),
EAMObjectRelationManagedBean.getTempPeriodTime(),
EAMObjectRelationManagedBean.newPeriodTime() und
EAMObjectRelationManagedBean.saveTime().

EAMPointTime tempPointTime [private]

Definiert in Zeile 60 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.editTime(),
EAMObjectRelationManagedBean.getTempPointTime(),
EAMObjectRelationManagedBean.newPointTime() und
EAMObjectRelationManagedBean.saveTime().

EAMStatus tempStatus [private]

Definiert in Zeile 58 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von `EAMObjectRelationManagedBean.checkDeleteStatus()`,
`EAMObjectRelationManagedBean.editStatus()`,
`EAMObjectRelationManagedBean.getNewStatus()`,
`EAMObjectRelationManagedBean.getTempStatus()`,
`EAMObjectRelationManagedBean.newStatus()`,
`EAMObjectRelationManagedBean.saveStatus()` und
`EAMObjectRelationManagedBean.setNewStatus()`.

EAMTime tempTime [private]

Definiert in Zeile 61 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.checkDeleteTime()` und
`EAMObjectRelationManagedBean.getTempTime()`.

EAMType tempType [private]

Definiert in Zeile 56 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.checkDeleteType()`,
`EAMObjectRelationManagedBean.editType()`,
`EAMObjectRelationManagedBean.getTempType()`,
`EAMObjectRelationManagedBean.newType()` und
`EAMObjectRelationManagedBean.saveType()`.

int typeCount = 0 [private]

Definiert in Zeile 93 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.checkDeleteType()`,
`EAMObjectRelationManagedBean.editType()`,
`EAMObjectRelationManagedBean.getPossibleTypes()`,
`EAMObjectRelationManagedBean.getTypeCount()` und
`EAMObjectRelationManagedBean.newType()`.

javax.faces.component.html.HtmlInputHidden typeCounter [private]

Definiert in Zeile 49 der Datei `EAMObjectRelationManagedBean.java`.

Wird benutzt von `EAMObjectRelationManagedBean.addType()`,
`EAMObjectRelationManagedBean.checkDeleteType()`,
`EAMObjectRelationManagedBean.editType()`,
`EAMObjectRelationManagedBean.getTypeCounter()` und
`EAMObjectRelationManagedBean.setTypeCounter()`.

String userId =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Definiert in Zeile 36 der Datei EAMObjectRelationManagedBean.java.

String year [private]

Definiert in Zeile 104 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.getYear(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newRelation() und
EAMObjectRelationManagedBean.saveTime().

String year2 [private]

Definiert in Zeile 105 der Datei EAMObjectRelationManagedBean.java.

Wird benutzt von EAMObjectRelationManagedBean.getYear2(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newRelation() und
EAMObjectRelationManagedBean.saveTime().

4.4.4 EAMStaticClass Klassenreferenz

4.4.4.1 Ausführliche Beschreibung

Statische Klasse gemeinsamen Zugriff aller ManagedBeans sowie Austausch von Objekten zwischen den ManagedBeans

Autor:

David Heimann
Roland Koppe

Definiert in Zeile 26 der Datei EAMStaticClass.java.

Öffentliche, statische Methoden

- static synchronized EAMStaticClass getInstance (String userId, String password, String clazz)

Öffentliche Attribute

- EAMAttributeDAO attributeDAOInterface
- EAMAttributeTypeDAO attributeTypeDAOInterface
- EAMCategoryDAO categoryDAOInterface
- EAMCategory currentCategory
- EAMSuperClass currentObjectRelation

- boolean newRequest
- EAMObjectDAO objectDAOInterface
- EAMObjectTypeDAO objectTypeDAOInterface
- EAMRelationDAO relationDAOInterface
- EAMRelationTypeDAO relationTypeDAOInterface
- EAMStatusDAO statusDAOInterface
- EAMTimeDAO timeDAOInterface
- EAMTypeDAO typeDAOInterface

Private Methoden

- EAMStaticClass (String userId, String password)
- EAMStaticClass ()
- EAMStaticClass reloadDAORights ()

Statische private Attribute

- static HashMap< String, EAMStaticClass > factories = new HashMap<String, EAMStaticClass>()
- static HashSet< String > registered = new HashSet<String>()

4.4.4.2 Beschreibung der Konstruktoren und Destruktoren

EAMStaticClass () [private]

Definiert in Zeile 65 der Datei EAMStaticClass.java.

Wird benutzt von EAMStaticClass.getInstance().

EAMStaticClass (String userId, String password) [private]

Definiert in Zeile 69 der Datei EAMStaticClass.java.

Benutzt EAMStaticClass.attributeDAOInterface, EAMStaticClass.attributeTypeDAOInterface, EAMStaticClass.categoryDAOInterface, EAMStaticClass.objectDAOInterface, EAMStaticClass.objectTypeDAOInterface, EAMStaticClass.relationDAOInterface, EAMStaticClass.relationTypeDAOInterface, EAMStaticClass.statusDAOInterface, EAMStaticClass.timeDAOInterface und EAMStaticClass.typeDAOInterface.

4.4.4.3 Dokumentation der Elementfunktionen

static synchronized EAMStaticClass getInstance (String userId, String password, String clazz) [static]

Gibt eine Instanz von EAMStaticClass eines Benutzers zurück

Definiert in Zeile 50 der Datei EAMStaticClass.java.

Benutzt EAMStaticClass.EAMStaticClass() und EAMStaticClass.factories.

EAMStaticClass reloadDAORights () [private]

Definiert in Zeile 83 der Datei EAMStaticClass.java.

Benutzt EAMStaticClass.attributeDAOInterface,
EAMStaticClass.attributeTypeDAOInterface, EAMStaticClass.categoryDAOInterface,
EAMStaticClass.objectDAOInterface, EAMStaticClass.objectTypeDAOInterface,
EAMStaticClass.relationDAOInterface, EAMStaticClass.relationTypeDAOInterface,
EAMStaticClass.statusDAOInterface, EAMStaticClass.timeDAOInterface und
EAMStaticClass.typeDAOInterface.

4.4.4.4 Dokumentation der Datenelemente

EAMAttributeDAO attributeDAOInterface

Definiert in Zeile 33 der Datei EAMStaticClass.java.

Wird benutzt von EAMStaticClass.EAMStaticClass(),
EAMAttributeManagedBean.getExistingAttributes() und
EAMStaticClass.reloadDAORights().

EAMAttributeTypeDAO attributeTypeDAOInterface

Definiert in Zeile 36 der Datei EAMStaticClass.java.

Wird benutzt von EAMAttributeManagedBean.checkDeleteType(),
EAMAttributeManagedBean.deleteAttributeType(), EAMStaticClass.EAMStaticClass(),
EAMAttributeManagedBean.editAttributeType(),
EAMAttributeManagedBean.getPossibleAttributeTypes(),
EAMStaticClass.reloadDAORights() und
EAMAttributeManagedBean.saveAttributeType().

EAMCategoryDAO categoryDAOInterface

Definiert in Zeile 32 der Datei EAMStaticClass.java.

Wird benutzt von EAMCategoryManagedBean.checkDelete(),
EAMCategoryManagedBean.deleteCategory(), EAMStaticClass.EAMStaticClass(),
EAMCategoryManagedBean.editCategory(),
EAMCategoryManagedBean.getCurrentEAMCategories(),
DataObjectManagedBean.openCategory(), EAMStaticClass.reloadDAORights() und
EAMCategoryManagedBean.saveCategory().

EAMCategory currentCategory

Definiert in Zeile 28 der Datei EAMStaticClass.java.

Wird benutzt von EAMObjectRelationManagedBean.getCurrentObjects(),
EAMObjectRelationManagedBean.getCurrentRelations(),
EAMObjectRelationManagedBean.getPossibleRelationsFromOwnMetamodel(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newRelation(),
EAMObjectRelationManagedBean.newType(),
EAMCategoryManagedBean.openCategory() und
EAMObjectRelationManagedBean.saveObjectRelation().

EAMSuperClass currentObjectRelation

Definiert in Zeile 29 der Datei EAMStaticClass.java.

Wird benutzt von EAMAttributeManagedBean.checkDeleteAttribute(),
EAMAttributeManagedBean.deleteAttribute(),
EAMAttributeManagedBean.editAttribute(),
EAMObjectRelationManagedBean.editObject(),
EAMObjectRelationManagedBean.editRelation(),
EAMAttributeManagedBean.getCurrentAttributes(),
EAMObjectRelationManagedBean.newObject(),
EAMObjectRelationManagedBean.newRelation(),
EAMAttributeManagedBean.saveAttribute() und
EAMObjectRelationManagedBean.saveObjectRelation().

HashMap<String, EAMStaticClass> factories = new HashMap<String, EAMStaticClass>() [static, private]

Definiert in Zeile 45 der Datei EAMStaticClass.java.

Wird benutzt von EAMStaticClass.getInstance().

boolean newRequest

Definiert in Zeile 42 der Datei EAMStaticClass.java.

Wird benutzt von EAMAttributeManagedBean.getCurrentAttributes(),
EAMObjectRelationManagedBean.getCurrentObjects(),
EAMObjectRelationManagedBean.getCurrentRelations(),
EAMObjectRelationManagedBean.getPossibleStatuses() und
EAMObjectRelationManagedBean.getPossibleTypes().

EAMObjectDAO objectDAOInterface

Definiert in Zeile 31 der Datei EAMStaticClass.java.

Wird benutzt von `EAMObjectRelationManagedBean.checkDeleteObject()`,
`EAMObjectRelationManagedBean.deleteObjectRelation()`,
`EAMStaticClass.EAMStaticClass()`, `EAMObjectRelationManagedBean.editObject()`,
`EAMObjectRelationManagedBean.getCurrentObjects()`,
`DataObjectManagedBean.getCurrentObjects()`,
`EAMObjectRelationManagedBean.getPossibleRelationsFromOtherMetamodels()`,
`EAMObjectRelationManagedBean.getPossibleRelationsFromOwnMetamodel()`,
`DataObjectManagedBean.pasteAll()`, `EAMStaticClass.reloadDAORights()`,
`EAMObjectRelationManagedBean.saveObjectRelation()` und
`DataObjectManagedBean.showDataObjectFromObject()`.

EAMObjectTypeDAO objectTypeDAOInterface

Definiert in Zeile 37 der Datei `EAMStaticClass.java`.

Wird benutzt von `EAMObjectRelationManagedBean.checkDeleteType()`,
`EAMObjectRelationManagedBean.deleteType()`, `EAMStaticClass.EAMStaticClass()`,
`EAMObjectRelationManagedBean.editType()`,
`EAMObjectRelationManagedBean.getPossibleTypes()`,
`EAMStaticClass.reloadDAORights()` und `EAMObjectRelationManagedBean.saveType()`.

`HashSet<String> registered = new HashSet<String>()` [static, private]

Definiert in Zeile 44 der Datei `EAMStaticClass.java`.

EAMRelationDAO relationDAOInterface

Definiert in Zeile 35 der Datei `EAMStaticClass.java`.

Wird benutzt von `EAMObjectRelationManagedBean.checkDeleteRelation()`,
`EAMObjectRelationManagedBean.deleteObjectRelation()`,
`EAMStaticClass.EAMStaticClass()`, `EAMObjectRelationManagedBean.editRelation()`,
`EAMObjectRelationManagedBean.getCurrentRelations()`,
`DataObjectManagedBean.getCurrentRelations()`, `DataObjectManagedBean.pasteAll()`,
`EAMStaticClass.reloadDAORights()`,
`EAMObjectRelationManagedBean.saveObjectRelation()` und
`DataObjectManagedBean.showDataObjectFromRelation()`.

EAMRelationTypeDAO relationTypeDAOInterface

Definiert in Zeile 38 der Datei `EAMStaticClass.java`.

Wird benutzt von `EAMObjectRelationManagedBean.checkDeleteType()`,
`EAMObjectRelationManagedBean.deleteType()`, `EAMStaticClass.EAMStaticClass()`,
`EAMObjectRelationManagedBean.editType()`,
`EAMObjectRelationManagedBean.getPossibleTypes()`,
`EAMStaticClass.reloadDAORights()` und `EAMObjectRelationManagedBean.saveType()`.

EAMStatusDAO statusDAOInterface

Definiert in Zeile 39 der Datei EAMStaticClass.java.

Wird benutzt von EAMObjectRelationManagedBean.checkDeleteStatus(),
EAMObjectRelationManagedBean.deleteStatus(), EAMStaticClass.EAMStaticClass(),
EAMObjectRelationManagedBean.editStatus(),
EAMObjectRelationManagedBean.getPossibleStatuses(),
EAMStaticClass.reloadDAORights() und EAMObjectRelationManagedBean.saveStatus().

EAMTimeDAO timeDAOInterface

Definiert in Zeile 40 der Datei EAMStaticClass.java.

Wird benutzt von EAMObjectRelationManagedBean.checkDeleteTime(),
EAMObjectRelationManagedBean.deleteTime(), EAMStaticClass.EAMStaticClass(),
EAMObjectRelationManagedBean.editTime(), EAMStaticClass.reloadDAORights() und
EAMObjectRelationManagedBean.saveTime().

EAMTypeDAO typeDAOInterface

Definiert in Zeile 34 der Datei EAMStaticClass.java.

Wird benutzt von EAMStaticClass.EAMStaticClass() und
EAMStaticClass.reloadDAORights().

5 de.offis.pg.eam.mod_extendDataInput

5.1 BundleConfiguration

5.1.1 BundleViews Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::module::api::IModuleView.

5.1.1.1 Ausführliche Beschreibung

Definiert in Zeile 21 der Datei BundleViews.java.

Öffentliche Methoden

- String getBundleVersion ()
- String getDescription ()
- long getEquinoxBundleId ()
- String getName ()
- List< View > getViews ()

Statische private Attribute

- static final String VIEWS = "bundle_views.xml"

5.1.1.2 Dokumentation der Elementfunktionen

String getBundleVersion ()

Definiert in Zeile 83 der Datei BundleViews.java.

String getDescription ()

Definiert in Zeile 28 der Datei BundleViews.java.

long getEquinoxBundleId ()

Definiert in Zeile 34 der Datei BundleViews.java.

String getName ()

Definiert in Zeile 40 der Datei BundleViews.java.

List<View> getViews ()

Definiert in Zeile 46 der Datei BundleViews.java.

Benutzt BundleViews.VIEWS.

5.1.1.3 Dokumentation der Datenelemente

final String VIEWS = "bundle_views.xml" [static, private]

Definiert in Zeile 24 der Datei BundleViews.java.

Wird benutzt von BundleViews.getViews().

5.1.2 Configuration Klassenreferenz

Abgeleitet von *de::offis::pg::eam::core::configuration::api::IBundleConfiguration*.

5.1.2.1 Ausführliche Beschreibung

Definiert in Zeile 10 der Datei Configuration.java.

Öffentliche Methoden

- boolean bundleHasObjects ()
- boolean bundleHasRelations ()
- boolean bundleHasViews ()
- Document getObjectsJdomDocument () throws JDOMException, IOException

5.1.2.2 Dokumentation der Elementfunktionen

boolean bundleHasObjects ()

Definiert in Zeile 13 der Datei Configuration.java.

boolean bundleHasRelations ()

Definiert in Zeile 28 der Datei Configuration.java.

boolean bundleHasViews ()

Definiert in Zeile 18 der Datei Configuration.java.

Document getObjectJdomDocument () throws JDOMException, IOException

Definiert in Zeile 23 der Datei Configuration.java.

5.1.3 Activator Klassenreferenz

Abgeleitet von org::osgi::framework::BundleActivator.

5.1.3.1 Ausführliche Beschreibung

Definiert in Zeile 38 der Datei Activator.java.

Öffentliche Methoden

- void start (BundleContext context) throws Exception
- void stop (BundleContext context) throws Exception

Statische öffentliche Attribute

- static String bundleSymbolicName
- static String bundleVersion
- static long equinoxBundleId
- static final String PATH = "/extendDataInput"

Private Attribute

- ServiceTracker httpServiceTracker

Statische private Attribute

- static final String WEB_ROOT = "/WebRoot"

Klassen

- class HttpServiceTracker
- class ServletContextListenerServletAdaptor

5.1.3.2 Dokumentation der Elementfunktionen**void start (BundleContext context) throws Exception**

Definiert in Zeile 51 der Datei Activator.java.

Benutzt `Activator.bundleSymbolicName`, `Activator.bundleVersion`, `Activator.equinoxBundleId` und `Activator.httpServiceTracker`.

`void stop (BundleContext context) throws Exception`

Definiert in Zeile 66 der Datei `Activator.java`.

Benutzt `Activator.httpServiceTracker`.

5.1.3.3 Dokumentation der Datenelemente

`String bundleSymbolicName [static]`

Definiert in Zeile 47 der Datei `Activator.java`.

Wird benutzt von `Activator.start()`.

`String bundleVersion [static]`

Definiert in Zeile 48 der Datei `Activator.java`.

Wird benutzt von `Activator.start()`.

`long equinoxBundleId [static]`

Definiert in Zeile 49 der Datei `Activator.java`.

Wird benutzt von `Activator.start()`.

`ServiceTracker httpServiceTracker [private]`

Definiert in Zeile 45 der Datei `Activator.java`.

Wird benutzt von `Activator.start()` und `Activator.stop()`.

`final String PATH = "/extendDataInput" [static]`

Definiert in Zeile 41 der Datei `Activator.java`.

Wird benutzt von `Activator.HttpServiceTracker.addingService()` und `Activator.HttpServiceTracker.removedService()`.

`final String WEB_ROOT = "/WebRoot" [static, private]`

Definiert in Zeile 43 der Datei `Activator.java`.

Wird benutzt von `Activator.HttpServiceTracker.addingService()`.

5.1.4 Activator.HttpServiceTracker Klassenreferenz

Abgeleitet von `org::osgi::util::tracker::ServiceTracker`.

5.1.4.1 Ausführliche Beschreibung

Definiert in Zeile 74 der Datei `Activator.java`.

Öffentliche Methoden

- `Object addingService (ServiceReference reference)`
- `HttpServiceTracker (BundleContext context)`
- `void removedService (ServiceReference reference, Object service)`

5.1.4.2 Beschreibung der Konstruktoren und Destruktoren

`HttpServiceTracker (BundleContext context)`

Definiert in Zeile 82 der Datei `Activator.java`.

5.1.4.3 Dokumentation der Elementfunktionen

`Object addingService (ServiceReference reference)`

Definiert in Zeile 90 der Datei `Activator.java`.

Benutzt `Activator.PATH` und `Activator.WEB_ROOT`.

`void removedService (ServiceReference reference, Object service)`

Definiert in Zeile 170 der Datei `Activator.java`.

Benutzt `Activator.PATH`.

5.1.5 Activator.ServletContextListenerServletAdaptor Klassenreferenz

Abgeleitet von `javax::servlet::Servlet`.

5.1.5.1 Ausführliche Beschreibung

Definiert in Zeile 188 der Datei `Activator.java`.

Öffentliche Methoden

- `void destroy ()`

- `ServletConfig getServletConfig ()`
- `String getServletInfo ()`
- `void init (ServletConfig config) throws ServletException`
- `void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException`
- `ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)`

Private Attribute

- `ServletConfig config`
- `Servlet delegate`
- `ClassLoader jspLoader`
- `ServletContextListener listener`

5.1.5.2 Beschreibung der Konstruktoren und Destruktoren

ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Definiert in Zeile 200 der Datei `Activator.java`.

5.1.5.3 Dokumentation der Elementfunktionen

void destroy ()

Definiert in Zeile 268 der Datei `Activator.java`.

Benutzt `Activator.ServletContextListenerServletAdaptor.config`, `Activator.ServletContextListenerServletAdaptor.delegate`, `Activator.ServletContextListenerServletAdaptor.jspLoader` und `Activator.ServletContextListenerServletAdaptor.listener`.

ServletConfig getServletConfig ()

Definiert in Zeile 296 der Datei `Activator.java`.

Benutzt `Activator.ServletContextListenerServletAdaptor.config`.

String getServletInfo ()

Definiert in Zeile 304 der Datei `Activator.java`.

void init (ServletConfig config) throws ServletException

Definiert in Zeile 216 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config,
Activator.ServletContextListenerServletAdaptor.delegate,
Activator.ServletContextListenerServletAdaptor.jspLoader und
Activator.ServletContextListenerServletAdaptor.listener.

void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException

Definiert in Zeile 244 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.delegate und
Activator.ServletContextListenerServletAdaptor.jspLoader.

5.1.5.4 Dokumentation der Datenelemente

ServletConfig config [private]

Definiert in Zeile 190 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.getServletConfig() und
Activator.ServletContextListenerServletAdaptor.init().

Servlet delegate [private]

Definiert in Zeile 194 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.init() und
Activator.ServletContextListenerServletAdaptor.service().

ClassLoader jspLoader [private]

Definiert in Zeile 196 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.init() und
Activator.ServletContextListenerServletAdaptor.service().

ServletContextListener listener [private]

Definiert in Zeile 192 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy() und
Activator.ServletContextListenerServletAdaptor.init().

5.1.6 MenuContributor Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::menu::api::AbstractAdoptableMenuContributor`.

5.1.6.1 Ausführliche Beschreibung

Definiert in Zeile 6 der Datei `MenuContributor.java`.

Öffentliche Methoden

- `String getAdopterId ()`
- `String getContributionTarget ()`
- `String getContributorId ()`
- `String getTargetSubmenu ()`
- `MenuContributor ()`

5.1.6.2 Beschreibung der Konstruktoren und Destruktoren

`MenuContributor ()`

Definiert in Zeile 8 der Datei `MenuContributor.java`.

5.1.6.3 Dokumentation der Elementfunktionen

`String getAdopterId ()`

Definiert in Zeile 12 der Datei `MenuContributor.java`.

`String getContributionTarget ()`

Definiert in Zeile 16 der Datei `MenuContributor.java`.

`String getContributorId ()`

Definiert in Zeile 20 der Datei `MenuContributor.java`.

`String getTargetSubmenu ()`

Definiert in Zeile 24 der Datei `MenuContributor.java`.

5.1.7 MenuConstants Schnittstellenreferenz

5.1.7.1 Ausführliche Beschreibung

Definiert in Zeile 3 der Datei MenuConstants.java.

Statische öffentliche Attribute

- static final String ADOPTER_ID = "extendDataInputAdopter"
- static final String CONTRIBUTOR_ID = "extendDataInputContributor"

5.1.7.2 Dokumentation der Datenelemente

final String ADOPTER_ID = "extendDataInputAdopter" [static]

Definiert in Zeile 7 der Datei MenuConstants.java.

final String CONTRIBUTOR_ID = "extendDataInputContributor" [static]

Definiert in Zeile 5 der Datei MenuConstants.java.

5.2 Beans

5.2.1 AttributeBean Klassenreferenz

5.2.1.1 Ausführliche Beschreibung

Autor:

Christian Zillmann Stellt ein Attribut einer Instanz dar

Definiert in Zeile 11 der Datei AttributeBean.java.

Öffentliche Methoden

- String getId ()
- String getName ()
- EAMAttributeType getType ()
- String getValue ()
- void setId (String id)
- void setName (String name)
- void setType (EAMAttributeType type)
- void setValue (String value)

Private Attribute

- String id
- String name
- EAMAttributeType type
- String value

5.2.1.2 Dokumentation der Elementfunktionen

String getId ()

Definiert in Zeile 45 der Datei AttributeBean.java.

Benutzt AttributeBean.id.

String getName ()

Definiert in Zeile 21 der Datei AttributeBean.java.

Benutzt AttributeBean.name.

EAMAttributeType getType ()

Definiert in Zeile 37 der Datei AttributeBean.java.

Benutzt AttributeBean.type.

String getValue ()

Definiert in Zeile 29 der Datei AttributeBean.java.

Benutzt AttributeBean.value.

void setId (String id)

Definiert in Zeile 49 der Datei AttributeBean.java.

Wird benutzt von RelationInstanceBean.getAttributes() und ObjectInstanceBean.getAttributes().

void setName (String name)

Definiert in Zeile 25 der Datei AttributeBean.java.

Wird benutzt von RelationInstanceBean.getAttributes() und ObjectInstanceBean.getAttributes().

void setType (EAMAttributeType type)

Definiert in Zeile 41 der Datei AttributeBean.java.

Wird benutzt von RelationInstanceBean.getAttributes() und ObjectInstanceBean.getAttributes().

void setValue (String value)

Definiert in Zeile 33 der Datei AttributeBean.java.

Wird benutzt von RelationInstanceBean.getAttributes() und ObjectInstanceBean.getAttributes().

5.2.1.3 Dokumentation der Datenelemente

String id [private]

Definiert in Zeile 17 der Datei AttributeBean.java.

Wird benutzt von AttributeBean.getId().

String name [private]

Definiert in Zeile 13 der Datei AttributeBean.java.

Wird benutzt von AttributeBean.getName().

EAMAttributeType type [private]

Definiert in Zeile 19 der Datei AttributeBean.java.

Wird benutzt von AttributeBean.getType().

String value [private]

Definiert in Zeile 15 der Datei AttributeBean.java.

Wird benutzt von AttributeBean.getValue().

5.2.2 InstanceBean Klassenreferenz

5.2.2.1 Ausführliche Beschreibung

Autor:

Christian Zillmann Stellt eine Instanz eines Objektes dar.

Definiert in Zeile 12 der Datei InstanceBean.java.

Öffentliche Methoden

- EAMCategory getCategory ()
- DataObject getDataobject ()
- EAMObject getEamobject ()
- String getId ()
- String getLabel ()
- boolean isLock ()
- void setCategory (EAMCategory category)
- void setDataobject (DataObject dataobject)
- void setEamobject (EAMObject eamobject)
- void setId (String id)
- void setLabel (String label)
- void setLock (boolean lock)

Private Attribute

- EAMCategory category
- DataObject dataobject
- EAMObject eamobject
- String id
- String label
- boolean lock

5.2.2.2 Dokumentation der Elementfunktionen

EAMCategory getCategory ()

Definiert in Zeile 67 der Datei InstanceBean.java.

Benutzt InstanceBean.category.

DataObject getDataobject ()

Definiert in Zeile 51 der Datei InstanceBean.java.

Benutzt InstanceBean.dataobject.

EAMObject getEamobject ()

Definiert in Zeile 59 der Datei InstanceBean.java.

Benutzt InstanceBean.eamobject.

String getId ()

Definiert in Zeile 43 der Datei InstanceBean.java.

Benutzt InstanceBean.id.

String getLabel ()

Definiert in Zeile 35 der Datei InstanceBean.java.

Benutzt InstanceBean.label.

Wird benutzt von ObjectInstancesBean.getObjectInstances().

boolean isLock ()

Definiert in Zeile 27 der Datei InstanceBean.java.

Benutzt InstanceBean.lock.

void setCategory (EAMCategory category)

Definiert in Zeile 71 der Datei InstanceBean.java.

Wird benutzt von ObjectInstancesBean.getObjectInstances(),
RelationInstancesBean.getRelationInstances() und
ObjectInstanceBean.getRelationInstances().

void setDataobject (DataObject dataobject)

Definiert in Zeile 55 der Datei InstanceBean.java.

Wird benutzt von ObjectInstancesBean.getObjectInstances().

void setEamobject (EAMObject eamobject)

Definiert in Zeile 63 der Datei InstanceBean.java.

Wird benutzt von ObjectInstancesBean.getObjectInstances(),
RelationInstancesBean.getRelationInstances() und
ObjectInstanceBean.getRelationInstances().

void setId (String id)

Definiert in Zeile 47 der Datei InstanceBean.java.

Wird benutzt von ObjectInstanceBean.getDetails(),
ObjectInstancesBean.getObjectInstances(), RelationInstancesBean.getRelationInstances()
und ObjectInstanceBean.getRelationInstances().

void setLabel (String label)

Definiert in Zeile 39 der Datei InstanceBean.java.

Wird benutzt von ObjectInstanceBean.getDetails(),
ObjectInstancesBean.getObjectInstances(), RelationInstancesBean.getRelationInstances()
und ObjectInstanceBean.getRelationInstances().

void setLock (boolean lock)

Definiert in Zeile 31 der Datei InstanceBean.java.

Wird benutzt von ObjectInstancesBean.getObjectInstances().

5.2.2.3 Dokumentation der Datenelemente

EAMCategory category [private]

Definiert in Zeile 18 der Datei InstanceBean.java.

Wird benutzt von InstanceBean.getCategory().

DataObject dataobject [private]

Definiert in Zeile 14 der Datei InstanceBean.java.

Wird benutzt von InstanceBean.getDataobject().

EAMObject eamobject [private]

Definiert in Zeile 16 der Datei InstanceBean.java.

Wird benutzt von InstanceBean.getEamobject().

String id [private]

Definiert in Zeile 20 der Datei InstanceBean.java.

Wird benutzt von InstanceBean.getId().

String label [private]

Definiert in Zeile 22 der Datei InstanceBean.java.

Wird benutzt von InstanceBean.getLabel().

boolean lock [private]

Definiert in Zeile 25 der Datei InstanceBean.java.

Wird benutzt von InstanceBean.isLock().

5.2.3 RelationBean Klassenreferenz

5.2.3.1 Ausführliche Beschreibung

Autor:

Christian Zillmann Stellt eine Relation dar.

Definiert in Zeile 9 der Datei RelationBean.java.

Öffentliche Methoden

- String getCategoryId ()
- String getCategoryName ()
- ArrayList< InstanceBean > getFirstMember ()
- String getId ()
- String getKardone ()
- String getKardtwo ()
- String getLabel ()
- String getMetaId ()
- ArrayList< InstanceBean > getSecondMember ()
- String getType ()
- String getType_first_member ()
- String getType_first_member_id ()
- String getType_second_member ()
- String getType_second_member_id ()
- boolean isLock ()
- RelationBean ()
- void setCategoryId (String categoryId)
- void setCategoryName (String categoryName)
- void setFirstMember (ArrayList< InstanceBean > firstMember)
- void setId (String id)
- void setKardone (String kardone)
- void setKardtwo (String kardtwo)
- void setLabel (String label)
- void setLock (boolean lock)
- void setMetaId (String metaId)
- void setSecondMember (ArrayList< InstanceBean > secondMember)
- void setType (String type)
- void setType_first_member (String type_first_member)
- void setType_first_member_id (String type_first_member_id)

- void setType_second_member (String type_second_member)
- void setType_second_member_id (String type_second_member_id)

Private Attribute

- String categoryId
- String categoryName
- ArrayList< InstanceBean > firstMember
- String id
- String kardone
- String kardtwo
- String label
- boolean lock
- String metaId
- ArrayList< InstanceBean > secondMember
- String type
- String type_first_member
- String type_first_member_id
- String type_second_member
- String type_second_member_id

5.2.3.2 Beschreibung der Konstruktoren und Destruktoren

RelationBean ()

Definiert in Zeile 98 der Datei RelationBean.java.

Benutzt RelationBean.firstMember und RelationBean.secondMember.

5.2.3.3 Dokumentation der Elementfunktionen

String getCategoryId ()

Definiert in Zeile 82 der Datei RelationBean.java.

Benutzt RelationBean.categoryId.

String getCategoryName ()

Definiert in Zeile 90 der Datei RelationBean.java.

Benutzt RelationBean.categoryName.

ArrayList<InstanceBean> getFirstMember ()

Definiert in Zeile 135 der Datei RelationBean.java.

Benutzt RelationBean.firstMember.

Wird benutzt von RelationInstancesBean.getRelationInstances() und ObjectInstanceBean.getRelationInstances().

String getId ()

Definiert in Zeile 103 der Datei RelationBean.java.

Benutzt RelationBean.id.

String getKardone ()

Definiert in Zeile 66 der Datei RelationBean.java.

Benutzt RelationBean.kardone.

String getKardtvo ()

Definiert in Zeile 74 der Datei RelationBean.java.

Benutzt RelationBean.kardtvo.

String getLabel ()

Definiert in Zeile 127 der Datei RelationBean.java.

Benutzt RelationBean.label.

String getMetaId ()

Definiert in Zeile 58 der Datei RelationBean.java.

Benutzt RelationBean.metaId.

ArrayList<InstanceBean> getSecondMember ()

Definiert in Zeile 143 der Datei RelationBean.java.

Benutzt RelationBean.secondMember.

Wird benutzt von RelationInstancesBean.getRelationInstances() und ObjectInstanceBean.getRelationInstances().

String getType ()

Definiert in Zeile 50 der Datei RelationBean.java.

Benutzt RelationBean.type.

Wird benutzt von RelationInstancesBean.getRelationInstances().

String getType_first_member ()

Definiert in Zeile 111 der Datei RelationBean.java.

Benutzt RelationBean.type_first_member.

String getType_first_member_id ()

Definiert in Zeile 151 der Datei RelationBean.java.

Benutzt RelationBean.type_first_member_id.

String getType_second_member ()

Definiert in Zeile 119 der Datei RelationBean.java.

Benutzt RelationBean.type_second_member.

String getType_second_member_id ()

Definiert in Zeile 159 der Datei RelationBean.java.

Benutzt RelationBean.type_second_member_id.

boolean isLock ()

Definiert in Zeile 42 der Datei RelationBean.java.

Benutzt RelationBean.lock.

void setCategoryId (String categoryId)

Definiert in Zeile 86 der Datei RelationBean.java.

Wird benutzt von RelationInstancesBean.getRelationInstances() und ObjectInstanceBean.getRelationInstances().

void setCategoryName (String categoryName)

Definiert in Zeile 94 der Datei RelationBean.java.

Wird benutzt von RelationInstancesBean.getRelationInstances() und ObjectInstanceBean.getRelationInstances().

void setFirstMember (ArrayList< InstanceBean > firstMember)

Definiert in Zeile 139 der Datei RelationBean.java.

void setId (String id)

Definiert in Zeile 107 der Datei RelationBean.java.

Wird benutzt von `RelationInstancesBean.getRelationInstances()` und `ObjectInstanceBean.getRelationInstances()`.

void setKardone (String kardone)

Definiert in Zeile 70 der Datei `RelationBean.java`.

Wird benutzt von `RelationInstancesBean.getRelationInstances()` und `ObjectInstanceBean.getRelationInstances()`.

void setKardtvo (String kardtvo)

Definiert in Zeile 78 der Datei `RelationBean.java`.

Wird benutzt von `RelationInstancesBean.getRelationInstances()` und `ObjectInstanceBean.getRelationInstances()`.

void setLabel (String label)

Definiert in Zeile 131 der Datei `RelationBean.java`.

Wird benutzt von `RelationInstancesBean.getRelationInstances()` und `ObjectInstanceBean.getRelationInstances()`.

void setLock (boolean lock)

Definiert in Zeile 46 der Datei `RelationBean.java`.

Wird benutzt von `RelationInstancesBean.getRelationInstances()`.

void setMetald (String metald)

Definiert in Zeile 62 der Datei `RelationBean.java`.

Wird benutzt von `RelationInstancesBean.getRelationInstances()` und `ObjectInstanceBean.getRelationInstances()`.

void setSecondMember (ArrayList< InstanceBean > secondMember)

Definiert in Zeile 147 der Datei `RelationBean.java`.

void setType (String type)

Definiert in Zeile 54 der Datei `RelationBean.java`.

Wird benutzt von `RelationInstancesBean.getRelationInstances()` und `ObjectInstanceBean.getRelationInstances()`.

void setType_first_member (String type_first_member)

Definiert in Zeile 115 der Datei RelationBean.java.

Wird benutzt von RelationInstancesBean.getRelationInstances() und ObjectInstanceBean.getRelationInstances().

void setType_first_member_id (String type_first_member_id)

Definiert in Zeile 155 der Datei RelationBean.java.

Wird benutzt von RelationInstancesBean.getRelationInstances() und ObjectInstanceBean.getRelationInstances().

void setType_second_member (String type_second_member)

Definiert in Zeile 123 der Datei RelationBean.java.

Wird benutzt von RelationInstancesBean.getRelationInstances() und ObjectInstanceBean.getRelationInstances().

void setType_second_member_id (String type_second_member_id)

Definiert in Zeile 163 der Datei RelationBean.java.

Wird benutzt von RelationInstancesBean.getRelationInstances() und ObjectInstanceBean.getRelationInstances().

5.2.3.4 Dokumentation der Datenelemente

String categoryId [private]

Definiert in Zeile 27 der Datei RelationBean.java.

Wird benutzt von RelationBean.getCategoryId().

String categoryName [private]

Definiert in Zeile 29 der Datei RelationBean.java.

Wird benutzt von RelationBean.getCategoryName().

ArrayList<InstanceBean> firstMember [private]

Definiert in Zeile 15 der Datei RelationBean.java.

Wird benutzt von RelationBean.getFirstMember() und RelationBean.RelationBean().

String id [private]

Definiert in Zeile 13 der Datei RelationBean.java.

Wird benutzt von `RelationBean.getId()`.

String kardone [private]

Definiert in Zeile 31 der Datei `RelationBean.java`.

Wird benutzt von `RelationBean.getKardone()`.

String kardtwo [private]

Definiert in Zeile 33 der Datei `RelationBean.java`.

Wird benutzt von `RelationBean.getKardtwo()`.

String label [private]

Definiert in Zeile 11 der Datei `RelationBean.java`.

Wird benutzt von `RelationBean.getLabel()`.

boolean lock [private]

Definiert in Zeile 39 der Datei `RelationBean.java`.

Wird benutzt von `RelationBean.isLock()`.

String metaId [private]

Definiert in Zeile 35 der Datei `RelationBean.java`.

Wird benutzt von `RelationBean.getMetaId()`.

ArrayList<InstanceBean> secondMember [private]

Definiert in Zeile 17 der Datei `RelationBean.java`.

Wird benutzt von `RelationBean.getSecondMember()` und `RelationBean.RelationBean()`.

String type [private]

Definiert in Zeile 37 der Datei `RelationBean.java`.

Wird benutzt von `RelationBean.getType()`.

String type_first_member [private]

Definiert in Zeile 23 der Datei `RelationBean.java`.

Wird benutzt von `RelationBean.getType_first_member()`.

String type_first_member_id [private]

Definiert in Zeile 19 der Datei RelationBean.java.

Wird benutzt von RelationBean.getType_first_member_id().

String type_second_member [private]

Definiert in Zeile 25 der Datei RelationBean.java.

Wird benutzt von RelationBean.getType_second_member().

String type_second_member_id [private]

Definiert in Zeile 21 der Datei RelationBean.java.

Wird benutzt von RelationBean.getType_second_member_id().

5.3 Managed Beans

5.3.1 ObjectInstanceBean Klassenreferenz

5.3.1.1 Ausführliche Beschreibung

Autor:

Christian Zillmann Diese Klasse enthält Methoden, um eine Instanz eines Metaobjektes zu betrachten, zu bearbeiten, zu löschen oder zu editieren.

Definiert in Zeile 56 der Datei ObjectInstanceBean.java.

Öffentliche Methoden

- String addAttribute ()
- String cancel ()
- String deleteAttribute ()
- String editAttribute ()
- ArrayList< AttributeBean > getAttributes ()
- String getCategoryId ()
- InstanceBean getDetails ()
- String getInstanceId ()
- String getMetamodellName ()
- String getMetaobjectId ()
- String getMetaobjectName ()
- String getNewAttributeName ()
- String getNewAttributeTypeField ()

- `ArrayList< RelationBean > getRelationInstances ()`
- `String makeCopy ()`
- `void setCategoryId (String categoryId)`
- `String setInstance ()`
- `void setInstanceId (String instanceId)`
- `void setMetaobjectId (String metaobjectId)`
- `void setNewAttributeName (String newAttributeName)`
- `void setNewAttributeTypeField (String newAttributeTypeField)`

Öffentliche Attribute

- `String password =`
`(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")`
- `String userId =`
`(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")`

Private Methoden

- `DataObject findDataObject (String meta, String id)`

Private Attribute

- `String categoryId`
- `EAMCategoryDAO eamCategoryDao`
- `EAMObjectDAO eamObjectDao`
- `EAMRelationDAO eamRelationDao`
- `String instanceId`
- `String metaobjectId`
- `String newAttributeName`
- `String newAttributeTypeField`

5.3.1.2 Dokumentation der Elementfunktionen

String addAttribute ()

Fügt ein Attribut der gesetzten Instanz hinzu

Rückgabe:

JSF-Workflow.

Definiert in Zeile 287 der Datei `ObjectInstanceBean.java`.

Benutzt `ObjectInstanceBean.eamObjectDao`, `ObjectInstanceBean.findDataObject()`, `ObjectInstanceBean.instanceId`, `ObjectInstanceBean.metaobjectId`, `ObjectInstanceBean.newAttributeName`, `ObjectInstanceBean.newAttributeTypeField` und `ObjectInstanceBean.password`.

String cancel ()

Definiert in Zeile 777 der Datei `ObjectInstanceBean.java`.

String deleteAttribute ()

Löscht ein Attribut der gesetzten Instanz.

Rückgabe:

JSF-WorkFlow

Definiert in Zeile 399 der Datei `ObjectInstanceBean.java`.

Benutzt `ObjectInstanceBean.findDataObject()`, `ObjectInstanceBean.instanceId`, `ObjectInstanceBean.metaobjectId` und `ObjectInstanceBean.password`.

String editAttribute ()

Editiert ein Attribut der gesetzten Instanz.

Rückgabe:

JSF-WorkFlow

Definiert in Zeile 622 der Datei `ObjectInstanceBean.java`.

Benutzt `ObjectInstanceBean.eamObjectDao`, `ObjectInstanceBean.instanceId`, `ObjectInstanceBean.password` und `ObjectInstanceBean.userId`.

DataObject findDataObject (String meta, String id) [private]

Mit dieser Methode kann man nach einen `DataObject` suchen.

Parameter:

meta Id des Metaobjektes

id Id des DataObjektes

Rückgabe:

`DataObject`

Definiert in Zeile 336 der Datei `ObjectInstanceBean.java`.

Benutzt `ObjectInstanceBean.eamObjectDao`, `ObjectInstanceBean.instanceId` und `ObjectInstanceBean.password`.

Wird benutzt von `ObjectInstanceBean.addAttribute()`,
`ObjectInstanceBean.deleteAttribute()`, `ObjectInstanceBean.getAttributes()` und
`ObjectInstanceBean.makeCopy()`.

`ArrayList<AttributeBean> getAttributes ()`

Liefert alle Attribute in Form von `AttributeBeans` in einer `ArrayList` zur gesetzten Instanz.

Rückgabe:

`ArrayList` mit `AttributeBeans`

Definiert in Zeile 245 der Datei `ObjectInstanceBean.java`.

Benutzt `ObjectInstanceBean.eamObjectDao`, `ObjectInstanceBean.findDataObject()`,
`ObjectInstanceBean.instanceId`, `ObjectInstanceBean.metaobjectId`, `AttributeBean.setId()`,
`AttributeBean.setName()`, `AttributeBean.setType()` und `AttributeBean.setValue()`.

`String getCategoryId ()`

Definiert in Zeile 387 der Datei `ObjectInstanceBean.java`.

Benutzt `ObjectInstanceBean.categoryId`.

`InstanceBean getDetails ()`

Liefert die Details zu der gesetzten Bean

Rückgabe:

Objekt vom Typ `InstanceBean` mit Details zur Instanz.

Definiert in Zeile 145 der Datei `ObjectInstanceBean.java`.

Benutzt `ObjectInstanceBean.categoryId`, `ObjectInstanceBean.eamObjectDao`,
`ObjectInstanceBean.instanceId`, `ObjectInstanceBean.metaobjectId`,
`ObjectInstanceBean.password`, `InstanceBean.setId()` und `InstanceBean.setLabel()`.

`String getInstanceId ()`

Definiert in Zeile 106 der Datei `ObjectInstanceBean.java`.

Benutzt `ObjectInstanceBean.instanceId`.

`String getMetamodelName ()`

Definiert in Zeile 418 der Datei `ObjectInstanceBean.java`.

Benutzt `ObjectInstanceBean.categoryId` und `ObjectInstanceBean.eamCategoryDao`.

`String getMetaobjectId ()`

Definiert in Zeile 98 der Datei ObjectInstanceBean.java.

Benutzt ObjectInstanceBean.metaobjectId.

String getMetaObjectName ()

Definiert in Zeile 426 der Datei ObjectInstanceBean.java.

Benutzt ObjectInstanceBean.eamObjectDao und ObjectInstanceBean.metaobjectId.

String getNewAttributeName ()

Definiert in Zeile 82 der Datei ObjectInstanceBean.java.

Benutzt ObjectInstanceBean.newAttributeName.

String getNewAttributeTypeField ()

Definiert in Zeile 90 der Datei ObjectInstanceBean.java.

Benutzt ObjectInstanceBean.newAttributeTypeField.

ArrayList<RelationBean> getRelationInstances ()

Liefert alle Relationen zurück, an der die gesetzte Instanz teilnimmt.

Rückgabe:

ArrayList mit RelationsBeans.

Definiert in Zeile 440 der Datei ObjectInstanceBean.java.

Benutzt ObjectInstanceBean.eamRelationDao, RelationBean.getFirstMember(), RelationBean.getSecondMember(), ObjectInstanceBean.instanceId, ObjectInstanceBean.metaobjectId, ObjectInstanceBean.password, InstanceBean.setCategory(), RelationBean.setCategoryId(), RelationBean.setCategoryName(), InstanceBean.setEamobject(), InstanceBean.setId(), RelationBean.setId(), RelationBean.setKardone(), RelationBean.setKardt看(), InstanceBean.setLabel(), RelationBean.setLabel(), RelationBean.setMetaId(), RelationBean.setType(), RelationBean.setType_first_member(), RelationBean.setType_first_member_id(), RelationBean.setType_second_member() und RelationBean.setType_second_member_id().

String makeCopy ()

Erstellt eine Kopie der gesetzten Instanz.

Rückgabe:

JSF-Workflow

Definiert in Zeile 784 der Datei ObjectInstanceBean.java.

Benutzt `ObjectInstanceBean.eamObjectDao`, `ObjectInstanceBean.findDataObject()`, `ObjectInstanceBean.instanceId`, `ObjectInstanceBean.metaobjectId` und `ObjectInstanceBean.password`.

void setCategoryId (String categoryId)

Definiert in Zeile 391 der Datei `ObjectInstanceBean.java`.

String setInstance ()

Setzt eine Instanz und zeigt diese dann an.

Rückgabe:

JSF-WorkFlow

Definiert in Zeile 119 der Datei `ObjectInstanceBean.java`.

Benutzt `ObjectInstanceBean.categoryId`, `ObjectInstanceBean.instanceId` und `ObjectInstanceBean.metaobjectId`.

void setInstanceId (String instanceId)

Definiert in Zeile 110 der Datei `ObjectInstanceBean.java`.

void setMetaobjectId (String metaobjectId)

Definiert in Zeile 102 der Datei `ObjectInstanceBean.java`.

void setNewAttributeName (String newAttributeName)

Definiert in Zeile 86 der Datei `ObjectInstanceBean.java`.

void setNewAttributeTypeField (String newAttributeTypeField)

Definiert in Zeile 94 der Datei `ObjectInstanceBean.java`.

5.3.1.3 Dokumentation der Datenelemente**String categoryId [private]**

Definiert in Zeile 76 der Datei `ObjectInstanceBean.java`.

Wird benutzt von `ObjectInstanceBean.getCategoryId()`, `ObjectInstanceBean.getDetails()`, `ObjectInstanceBean.getMetamodelName()` und `ObjectInstanceBean.setInstance()`.

EAMCategoryDAO eamCategoryDao [private]**Initialisierung:**

```
DAOFactory.getAuthInstance(userId, password)
            .getEAMCategoryDAO()
```

Definiert in Zeile 66 der Datei `ObjectInstanceBean.java`.

Wird benutzt von `ObjectInstanceBean.getMetamodelName()`.

EAMObjectDAO `eamObjectDao` [private]

Initialisierung:

```
DAOFactory.getAuthInstance(userId, password)
            .getEAMObjectDAO()
```

Definiert in Zeile 63 der Datei `ObjectInstanceBean.java`.

Wird benutzt von `ObjectInstanceBean.addAttribute()`,
`ObjectInstanceBean.editAttribute()`, `ObjectInstanceBean.findDataObject()`,
`ObjectInstanceBean.getAttributes()`, `ObjectInstanceBean.getDetails()`,
`ObjectInstanceBean.getMetaobjectName()` und `ObjectInstanceBean.makeCopy()`.

EAMRelationDAO `eamRelationDao` [private]

Initialisierung:

```
DAOFactory.getAuthInstance(userId, password)
            .getEAMRelationDAO()
```

Definiert in Zeile 69 der Datei `ObjectInstanceBean.java`.

Wird benutzt von `ObjectInstanceBean.getRelationInstances()`.

String `instanceId` [private]

Definiert in Zeile 74 der Datei `ObjectInstanceBean.java`.

Wird benutzt von `ObjectInstanceBean.addAttribute()`,
`ObjectInstanceBean.deleteAttribute()`, `ObjectInstanceBean.editAttribute()`,
`ObjectInstanceBean.findDataObject()`, `ObjectInstanceBean.getAttributes()`,
`ObjectInstanceBean.getDetails()`, `ObjectInstanceBean.getInstanceId()`,
`ObjectInstanceBean.getRelationInstances()`, `ObjectInstanceBean.makeCopy()` und
`ObjectInstanceBean.setInstance()`.

String `metaobjectId` [private]

Definiert in Zeile 72 der Datei `ObjectInstanceBean.java`.

Wird benutzt von `ObjectInstanceBean.addAttribute()`,
`ObjectInstanceBean.deleteAttribute()`, `ObjectInstanceBean.getAttributes()`,

`ObjectInstanceBean.getDetails()`, `ObjectInstanceBean.getMetaobjectId()`,
`ObjectInstanceBean.getMetaobjectName()`, `ObjectInstanceBean.getRelationInstances()`,
`ObjectInstanceBean.makeCopy()` und `ObjectInstanceBean.setInstance()`.

String newAttributeName [private]

Definiert in Zeile 80 der Datei `ObjectInstanceBean.java`.

Wird benutzt von `ObjectInstanceBean.addAttribute()` und
`ObjectInstanceBean.getNewAttributeName()`.

String newAttributeTypeField [private]

Definiert in Zeile 78 der Datei `ObjectInstanceBean.java`.

Wird benutzt von `ObjectInstanceBean.addAttribute()` und
`ObjectInstanceBean.getNewAttributeTypeField()`.

String password =

`(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")`

Definiert in Zeile 60 der Datei `ObjectInstanceBean.java`.

Wird benutzt von `ObjectInstanceBean.addAttribute()`,
`ObjectInstanceBean.deleteAttribute()`, `ObjectInstanceBean.editAttribute()`,
`ObjectInstanceBean.findDataObject()`, `ObjectInstanceBean.getDetails()`,
`ObjectInstanceBean.getRelationInstances()` und `ObjectInstanceBean.makeCopy()`.

String userId =

`(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")`

Definiert in Zeile 59 der Datei `ObjectInstanceBean.java`.

Wird benutzt von `ObjectInstanceBean.editAttribute()`.

5.3.2 ObjectInstancesBean Klassenreferenz

5.3.2.1 Ausführliche Beschreibung

Autor:

Christian Zillmann Diese Klasse dient zum Auflisten von Instanzen von Metaobjekten.

Definiert in Zeile 49 der Datei `ObjectInstancesBean.java`.

Öffentliche Methoden

- String addMetamodelFilter ()
- String addMetaobjectFilter ()
- String cancel ()
- String changeMetaObjectFunction ()
- String deleteInstance ()
- javax.faces.component.html.HtmlCommandButton getChangeMetaObject ()
- String getMetamodelFilter ()
- String getMetamodelFilterFromSession ()
- Long getMetamodelFilterIdFromSession ()
- List< SelectItem > getMetaModelsForSelectOne ()
- String getMetaobjectFilter ()
- String getMetaobjectFilterFromSession ()
- javax.faces.component.html.HtmlSelectOneMenu getMetaObjectMenu ()
- List< SelectItem > getMetaObjectsForSelectOne ()
- String getNewCategoryId ()
- List< InstanceBean > getObjectInstances ()
- javax.faces.component.html.HtmlCommandButton getSaveInstanceButton ()
- String getSearchField ()
- String getSearchFilterFromSession ()
- javax.faces.component.html.HtmlCommandButton getSetMetaObject ()
- boolean isAlreadySetMetaObject ()
- String makeInstance ()
- String removeMetamodelFilter ()
- String removeMetaobjectFilter ()
- void removeSearchFilter ()
- void search ()
- void setChangeMetaObject (javax.faces.component.html.HtmlCommandButton changeMetaObject)
- void setMetamodelFilter (String metamodelFilter)
- void setMetaobjectFilter (String metaobjectFilter)
- String setMetaObjectFunction ()
- void setMetaObjectMenu (javax.faces.component.html.HtmlSelectOneMenu metaObjectMenu)
- void setNewCategoryId (String newCategoryId)
- void setSaveInstanceButton (javax.faces.component.html.HtmlCommandButton saveInstanceButton)
- void setSearchField (String searchField)

- void setSetMetaObject (javax.faces.component.html.HtmlCommandButton setMetaObject)

Öffentliche Attribute

- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Private Methoden

- DataObject findDataObject (String meta, String id)

Private Attribute

- javax.faces.component.html.HtmlCommandButton changeMetaObject
- EAMCategoryDAO eamCategoryDao
- EAMObjectDAO eamObjectDao
- String metamodelFilter
- String metaobjectFilter
- javax.faces.component.html.HtmlSelectOneMenu metaObjectMenu
- String newCategoryId
- javax.faces.component.html.HtmlCommandButton saveInstanceButton
- String searchField
- javax.faces.component.html.HtmlCommandButton setMetaObject

5.3.2.2 Dokumentation der Elementfunktionen

String addMetamodelFilter ()

Mit dieser Methode wird der Metamodell-Filter gesetzt.

Autor:

christian zillmann

Rückgabe:

String für den Workflow von JSF.

Definiert in Zeile 398 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.metamodelFilter.

Wird benutzt von `ObjectInstancesBean.getMetamodelFilterFromSession()` und `ObjectInstancesBean.getObjectInstances()`.

String addMetaobjectFilter ()

Mit dieser Methode wird der Metaobjekt-Filter gesetzt.

Autor:

christian zillmann

Rückgabe:

String für den Workflow von JSF.

Definiert in Zeile 416 der Datei `ObjectInstancesBean.java`.

Benutzt `ObjectInstancesBean.metaobjectFilter`.

String cancel ()

Definiert in Zeile 716 der Datei `ObjectInstancesBean.java`.

Benutzt `ObjectInstancesBean.changeMetaObject`, `ObjectInstancesBean.metaObjectMenu`, `ObjectInstancesBean.saveInstanceButton` und `ObjectInstancesBean.setMetaObject`.

String changeMetaObjectFunction ()

Die Funktion wird aufgerufen, wenn das MetaObjekt, von dem eine neue Instanz erzeugt werden soll, bereits ausgewählt wurde aber noch einmal geändert werden soll.

Rückgabe:

String für den JSF Workflow

Definiert in Zeile 705 der Datei `ObjectInstancesBean.java`.

Benutzt `ObjectInstancesBean.changeMetaObject`, `ObjectInstancesBean.metaObjectMenu`, `ObjectInstancesBean.saveInstanceButton` und `ObjectInstancesBean.setMetaObject`.

String deleteInstance ()

Löscht eine Instanz.

Autor:

christian zillmann

Rückgabe:

String für den JSF Workflow

Definiert in Zeile 566 der Datei `ObjectInstancesBean.java`.

Benutzt `ObjectInstancesBean.findDataObject()` und `ObjectInstancesBean.password`.

DataObject findDataObject (String meta, String id) [private]

Diese Methode findet die Instanz zu einer entsprechenden Id und die Id ihres Metaobjekts.

Parameter:

meta

id

Rückgabe:

DataObject

Definiert in Zeile 613 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.eamObjectDao und ObjectInstancesBean.password.

Wird benutzt von ObjectInstancesBean.deleteInstance().

javax.faces.component.html.HtmlCommandButton getChangeMetaObject ()

Definiert in Zeile 735 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.changeMetaObject.

String getMetamodelFilter ()

Definiert in Zeile 778 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.metamodelFilter.

String getMetamodelFilterFromSession ()

Diese Methode holt den Metamodell-Filter der gerade gesetzt ist, aus der Session.

Autor:

christian zillmann

Rückgabe:

Name des Metamodells

Definiert in Zeile 480 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.addMetamodelFilter(),

ObjectInstancesBean.eamCategoryDao,

ObjectInstancesBean.getMetamodelFilterIdFromSession() und

ObjectInstancesBean.metamodelFilter.

Long getMetamodelFilterIdFromSession ()

Diese Methode holt den Metamodell-Filter der gerade gesetzt ist, aus der Session.

Autor:

christian zillmann

Rückgabe:

Name des Metamodells

Definiert in Zeile 512 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.eamCategoryDao und
ObjectInstancesBean.metamodelFilter.

Wird benutzt von ObjectInstancesBean.getMetamodelFilterFromSession() und
ObjectInstancesBean.getObjectInstances().

List<SelectItem> getMetaModelsForSelectOne ()

Diese Methode liefert eine Liste von SelectItems mit den gesamtem Metamodellen zurueck.

Autor:

christian zillmann

Rückgabe:

Liste mit Metamodellen als SelectItems.

Definiert in Zeile 114 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.eamCategoryDao.

String getMetaobjectFilter ()

Definiert in Zeile 782 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.metaobjectFilter.

String getMetaobjectFilterFromSession ()

Diese Methode holt den Metaobjekt-Filter der gerade gesetzt ist, aus der Session.

Autor:

christian zillmann

Rückgabe:

Name des Metaobjekts

Definiert in Zeile 459 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.eamObjectDao und ObjectInstancesBean.metaobjectFilter.

javax.faces.component.html.HtmlSelectOneMenu getMetaObjectMenu ()

Definiert in Zeile 744 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.metaObjectMenu.

List<SelectedItem> getMetaObjectsForSelectOne ()

Diese Methode baut eine Liste von SelectItems von Metaobjekten gruppiert nach ihrem Methamodell zusammen, damit man diese dann in einer SelectOneBox ausgeben kann

Autor:

christian zillmann

Rückgabe:

List von SelectItems gruppiert nach Metamodell

Definiert in Zeile 83 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.eamCategoryDao, ObjectInstancesBean.eamObjectDao und ObjectInstancesBean.metamodelFilter.

String getNewCategoryId ()

Definiert in Zeile 267 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.newCategoryId.

List<InstanceBean> getObjectInstances ()

Liefer alle Instanzen von Objekten zurück, auf eventuelle Filter, die gesetzt sind, wird geachtet.

Autor:

christian zillmann

Rückgabe:

Liste mit Instanzen von Metaobjekten.

Definiert in Zeile 282 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.addMetamodelFilter(), ObjectInstancesBean.eamCategoryDao, ObjectInstancesBean.eamObjectDao, InstanceBean.getLabel(), ObjectInstancesBean.getMetamodelFilterIdFromSession(), ObjectInstancesBean.metamodelFilter, ObjectInstancesBean.metaobjectFilter, ObjectInstancesBean.password, InstanceBean.setCategory(), InstanceBean.setDataobject(), InstanceBean.setEamobject(), InstanceBean.setId(), InstanceBean.setLabel() und InstanceBean.setLock().

javax.faces.component.html.HtmlCommandButton getSaveInstanceButton ()

Definiert in Zeile 753 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.saveInstanceButton.

String getSearchField ()

Definiert in Zeile 762 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.searchField.

String getSearchFilterFromSession ()

Diese Methode holt den Such-Filter der gerade gesetzt ist, aus der Session.

Autor:

christian zillmann

Rückgabe:

Suchwort

Definiert in Zeile 536 der Datei ObjectInstancesBean.java.

javax.faces.component.html.HtmlCommandButton getSetMetaObject ()

Definiert in Zeile 726 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.setMetaObject.

boolean isAlreadySetMetaObject ()

Diese Methode überprüft, ob bereits ein MetaObjekt ausgewählt wurde, von dem die Instanz erzeugt wurde.

Rückgabe:

true wenn bereits ausgewählt ansonsten false.

Definiert in Zeile 690 der Datei ObjectInstancesBean.java.

String makeInstance ()

Diese Methode legt eine neue Instanz an.

Autor:

christian zillmann

Rückgabe:

String für den Workflow von JSF.

Definiert in Zeile 130 der Datei ObjectInstancesBean.java.

Benutzt `ObjectInstancesBean.changeMetaObject`, `ObjectInstancesBean.eamObjectDao`, `ObjectInstancesBean.metaObjectMenu`, `ObjectInstancesBean.newCategoryId`, `ObjectInstancesBean.password`, `ObjectInstancesBean.saveInstanceButton` und `ObjectInstancesBean.setMetaObject`.

String removeMetamodelFilter ()

Mit dieser Methode wird der Metamodell-Filter entfernt.

Autor:

christian zillmann

Rückgabe:

String für den Workflow von JSF.

Definiert in Zeile 430 der Datei `ObjectInstancesBean.java`.

String removeMetaobjectFilter ()

Mit dieser Methode wird der Metamobjekt-Filter entfernt.

Autor:

christian zillmann

Rückgabe:

String für den Workflow von JSF.

Definiert in Zeile 444 der Datei `ObjectInstancesBean.java`.

void removeSearchFilter ()

Definiert in Zeile 546 der Datei `ObjectInstancesBean.java`.

void search ()

Definiert in Zeile 553 der Datei `ObjectInstancesBean.java`.

Benutzt `ObjectInstancesBean.searchField`.

void setChangeMetaObject (javax.faces.component.html.HtmlCommandButton changeMetaObject)

Definiert in Zeile 739 der Datei `ObjectInstancesBean.java`.

Benutzt `ObjectInstancesBean.changeMetaObject`.

void setMetamodelFilter (String metamodelFilter)

Definiert in Zeile 774 der Datei `ObjectInstancesBean.java`.

void setMetaobjectFilter (String metaobjectFilter)

Definiert in Zeile 770 der Datei ObjectInstancesBean.java.

String setMetaObjectFunction ()

Diese Methode setzt, das MetaObjekt kurzfristig in die Session, von dem eine Instanz erzeugt werden soll.

Autor:

christian zillmann

Rückgabe:

String für den JSF-Workflow

Definiert in Zeile 674 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.changeMetaObject, ObjectInstancesBean.metaObjectMenu, ObjectInstancesBean.newCategoryId, ObjectInstancesBean.saveInstanceButton und ObjectInstancesBean.setMetaObject.

void setMetaObjectMenu (javax.faces.component.html.HtmlSelectOneMenu metaObjectMenu)

Definiert in Zeile 748 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.metaObjectMenu.

void setNewCategoryId (String newCategoryId)

Definiert in Zeile 271 der Datei ObjectInstancesBean.java.

void setSaveInstanceButton (javax.faces.component.html.HtmlCommandButton saveInstanceButton)

Definiert in Zeile 757 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.saveInstanceButton.

void setSearchField (String searchField)

Definiert in Zeile 766 der Datei ObjectInstancesBean.java.

void setSetMetaObject (javax.faces.component.html.HtmlCommandButton setMetaObject)

Definiert in Zeile 730 der Datei ObjectInstancesBean.java.

Benutzt ObjectInstancesBean.setMetaObject.

5.3.2.3 Dokumentation der Datenelemente

javax.faces.component.html.HtmlCommandButton changeMetaObject [private]

Definiert in Zeile 69 der Datei ObjectInstancesBean.java.

Wird benutzt von ObjectInstancesBean.chancel(),
ObjectInstancesBean.changeMetaObjectFunction(),
ObjectInstancesBean.getChangeMetaObject(), ObjectInstancesBean.makeInstance(),
ObjectInstancesBean.setChangeMetaObject() und
ObjectInstancesBean.setMetaObjectFunction().

EAMCategoryDAO eamCategoryDao [private]

Initialisierung:

```
DAOFactory.getAuthInstance(userId, password)
            .getEAMCategoryDAO()
```

Definiert in Zeile 56 der Datei ObjectInstancesBean.java.

Wird benutzt von ObjectInstancesBean.getMetamodelFilterFromSession(),
ObjectInstancesBean.getMetamodelFilterIdFromSession(),
ObjectInstancesBean.getMetaModelsForSelectOne(),
ObjectInstancesBean.getMetaObjectsForSelectOne() und
ObjectInstancesBean.getObjectInstances().

EAMObjectDAO eamObjectDao [private]

Initialisierung:

```
DAOFactory.getAuthInstance(userId, password)
            .getEAMObjectDAO()
```

Definiert in Zeile 54 der Datei ObjectInstancesBean.java.

Wird benutzt von ObjectInstancesBean.findDataObject(),
ObjectInstancesBean.getMetaobjectFilterFromSession(),
ObjectInstancesBean.getMetaObjectsForSelectOne(),
ObjectInstancesBean.getObjectInstances() und ObjectInstancesBean.makeInstance().

String metamodelFilter [private]

Definiert in Zeile 61 der Datei ObjectInstancesBean.java.

Wird benutzt von ObjectInstancesBean.addMetamodelFilter(),
ObjectInstancesBean.getMetamodelFilter(),
ObjectInstancesBean.getMetamodelFilterFromSession(),

ObjectInstancesBean.getMetamodelFilterIdFromSession(),
ObjectInstancesBean.getMetaObjectsForSelectOne() und
ObjectInstancesBean.getObjectInstances().

String metaobjectFilter [private]

Definiert in Zeile 63 der Datei ObjectInstancesBean.java.

Wird benutzt von ObjectInstancesBean.addMetaobjectFilter(),
ObjectInstancesBean.getMetaobjectFilter(),
ObjectInstancesBean.getMetaobjectFilterFromSession() und
ObjectInstancesBean.getObjectInstances().

javax.faces.component.html.HtmlSelectOneMenu metaObjectMenu [private]

Definiert in Zeile 71 der Datei ObjectInstancesBean.java.

Wird benutzt von ObjectInstancesBean.chancel(),
ObjectInstancesBean.changeMetaObjectFunction(),
ObjectInstancesBean.getMetaObjectMenu(), ObjectInstancesBean.makeInstance(),
ObjectInstancesBean.setMetaObjectFunction() und
ObjectInstancesBean.setMetaObjectMenu().

String newCategoryId [private]

Definiert in Zeile 59 der Datei ObjectInstancesBean.java.

Wird benutzt von ObjectInstancesBean.getNewCategoryId(),
ObjectInstancesBean.makeInstance() und ObjectInstancesBean.setMetaObjectFunction().

String password =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")

Definiert in Zeile 52 der Datei ObjectInstancesBean.java.

Wird benutzt von ObjectInstancesBean.deleteInstance(),
ObjectInstancesBean.findDataObject(), ObjectInstancesBean.getObjectInstances() und
ObjectInstancesBean.makeInstance().

javax.faces.component.html.HtmlCommandButton saveInstanceButton [private]

Definiert in Zeile 73 der Datei ObjectInstancesBean.java.

Wird benutzt von ObjectInstancesBean.chancel(),
ObjectInstancesBean.changeMetaObjectFunction(),
ObjectInstancesBean.getSaveInstanceButton(), ObjectInstancesBean.makeInstance(),

`ObjectInstancesBean.setMetaObjectFunction()` und
`ObjectInstancesBean.setSaveInstanceButton()`.

String searchField [private]

Definiert in Zeile 65 der Datei `ObjectInstancesBean.java`.

Wird benutzt von `ObjectInstancesBean.getSearchField()` und
`ObjectInstancesBean.search()`.

javax.faces.component.html.HtmlCommandButton setMetaObject [private]

Definiert in Zeile 67 der Datei `ObjectInstancesBean.java`.

Wird benutzt von `ObjectInstancesBean.chancel()`,
`ObjectInstancesBean.changeMetaObjectFunction()`,
`ObjectInstancesBean.getSetMetaObject()`, `ObjectInstancesBean.makeInstance()`,
`ObjectInstancesBean.setMetaObjectFunction()` und
`ObjectInstancesBean.setSetMetaObject()`.

String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Definiert in Zeile 51 der Datei `ObjectInstancesBean.java`.

5.3.3 RelationInstanceBean Klassenreferenz

5.3.3.1 Ausführliche Beschreibung

Autor:

Christian Zillmann Diese Klasse enthält Methoden, um eine Instanz einer Metarelation zu betrachten, zu bearbeiten, zu löschen oder zu editieren.

Definiert in Zeile 51 der Datei `RelationInstanceBean.java`.

Öffentliche Methoden

- `String chancel ()`
- `String editAttribute ()`
- `ArrayList< AttributeBean > getAttributes ()`
- `String getCategory ()`
- `String getCategoryId ()`
- `String getFirstKard ()`
- `ArrayList< DataObject > getFirstMember ()`

- String getFirstmembertyp ()
- String getInstanceId ()
- String getLabel ()
- String getMetarelationId ()
- String getSecondKard ()
- ArrayList< DataObject > getSecondMember ()
- String getSecondmembertyp ()
- String getType ()
- RelationInstanceBean ()
- void setCategoryId (String categoryId)
- String setInstance ()
- void setInstanceId (String instanceId)
- void setMetarelationId (String metarelationId)

Öffentliche Attribute

- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Private Methoden

- DataObject findDataObject (String meta, String id)

Private Attribute

- String categoryId
- DataObject d
- EAMCategoryDAO eamCategoryDao
- EAMObjectDAO eamObjectDao
- EAMRelationDAO eamRelationDao
- String instanceId
- String metarelationId
- EAMRelation r

5.3.3.2 Beschreibung der Konstruktoren und Destruktoren

RelationInstanceBean ()

Definiert in Zeile 56 der Datei RelationInstanceBean.java.

5.3.3.3 Dokumentation der Elementfunktionen

String cancel ()

Definiert in Zeile 581 der Datei RelationInstanceBean.java.

String editAttribute ()

Editiert ein Attribute

Rückgabe:

JSF-Workflow

Definiert in Zeile 403 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.eamRelationDao, RelationInstanceBean.instanceId, RelationInstanceBean.password und RelationInstanceBean.userId.

DataObject findDataObject (String meta, String id) [private]

Sucht ein DataObject

Parameter:

meta Id des Metaobjektes

id Id der Instanz

Rückgabe:

DataObject

Definiert in Zeile 258 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.d, RelationInstanceBean.eamRelationDao, RelationInstanceBean.instanceId und RelationInstanceBean.password.

Wird benutzt von RelationInstanceBean.getAttributes(), RelationInstanceBean.getType() und RelationInstanceBean.setInstance().

ArrayList<AttributeBean> getAttributes ()

Liefert eine ArrayList mit den Attributen der gesetzten Instanz.

Rückgabe:

ArrayList<AttributeBean>

Definiert in Zeile 317 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.eamRelationDao, RelationInstanceBean.findDataObject(), RelationInstanceBean.instanceId, RelationInstanceBean.metarelationId,

AttributeBean.setId(), AttributeBean.setName(), AttributeBean.setType() und AttributeBean.setValue().

String getCategory ()

Definiert in Zeile 114 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.d.

String getCategoryId ()

Definiert in Zeile 391 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.categoryId.

String getFirstKard ()

Definiert in Zeile 126 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.r.

ArrayList<DataObject> getFirstMember ()

Liefert die Firstmember zu der gesetzten Instanz.

Rückgabe:

ArrayList<DataObject>

Definiert in Zeile 138 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.d, RelationInstanceBean.password und RelationInstanceBean.r.

String getFirstmembertyp ()

Definiert in Zeile 118 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.r.

String getInstanceId ()

Definiert in Zeile 573 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.instanceId.

String getLabel ()

Definiert in Zeile 110 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.d.

String getMetarelationId ()

Definiert in Zeile 565 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.metarelationId.

String getSecondKard ()

Definiert in Zeile 130 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.r.

ArrayList<DataObject> getSecondMember ()

Liefert die Secondmember zu der gesetzten Instanz.

Rückgabe:

ArrayList<DataObject>

Definiert in Zeile 199 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.d, RelationInstanceBean.password und RelationInstanceBean.r.

String getSecondmembertyp ()

Definiert in Zeile 122 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.r.

String getType ()

Liefert die Metarelation zu der gesetzten Instanz.

Rückgabe:

Name der Metarelation

Definiert in Zeile 355 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.categoryId, RelationInstanceBean.d, RelationInstanceBean.eamRelationDao, RelationInstanceBean.findDataObject(), RelationInstanceBean.instanceId, RelationInstanceBean.metarelationId und RelationInstanceBean.r.

void setCategoryId (String categoryId)

Definiert in Zeile 395 der Datei RelationInstanceBean.java.

String setInstance ()

Setzt eine Instanz, die betrachtet werden soll.

Rückgabe:

JSF-Workflow

Definiert in Zeile 84 der Datei RelationInstanceBean.java.

Benutzt RelationInstanceBean.categoryId, RelationInstanceBean.d, RelationInstanceBean.eamRelationDao, RelationInstanceBean.findDataObject(), RelationInstanceBean.instanceId, RelationInstanceBean.metarelationId und RelationInstanceBean.r.

void setInstancelId (String instancelId)

Definiert in Zeile 577 der Datei RelationInstanceBean.java.

void setMetarelationId (String metarelationId)

Definiert in Zeile 569 der Datei RelationInstanceBean.java.

5.3.3.4 Dokumentation der Datenelemente

String categoryId [private]

Definiert in Zeile 73 der Datei RelationInstanceBean.java.

Wird benutzt von RelationInstanceBean.getCategoryId(), RelationInstanceBean.getType() und RelationInstanceBean.setInstance().

DataObject d [private]

Definiert in Zeile 75 der Datei RelationInstanceBean.java.

Wird benutzt von RelationInstanceBean.findDataObject(), RelationInstanceBean.getCategory(), RelationInstanceBean.getFirstMember(), RelationInstanceBean.getLabel(), RelationInstanceBean.getSecondMember(), RelationInstanceBean.getType() und RelationInstanceBean.setInstance().

EAMCategoryDAO eamCategoryDao [private]

Initialisierung:

```
DAOFactory.getAuthInstance(userId, password)
            .getEAMCategoryDAO()
```

Definiert in Zeile 63 der Datei RelationInstanceBean.java.

EAMObjectDAO eamObjectDao [private]

Initialisierung:

```
DAOFactory.getAuthInstance(userId, password)
            .getEAMObjectDAO()
```

Definiert in Zeile 60 der Datei RelationInstanceBean.java.

EAMRelationDAO eamRelationDao [private]

Initialisierung:

```
DAOFactory.getAuthInstance(userId, password)
            .getEAMRelationDAO()
```

Definiert in Zeile 66 der Datei RelationInstanceBean.java.

Wird benutzt von RelationInstanceBean.editAttribute(),
RelationInstanceBean.findDataObject(), RelationInstanceBean.getAttributes(),
RelationInstanceBean.getType() und RelationInstanceBean.setInstance().

String instanceId [private]

Definiert in Zeile 71 der Datei RelationInstanceBean.java.

Wird benutzt von RelationInstanceBean.editAttribute(),
RelationInstanceBean.findDataObject(), RelationInstanceBean.getAttributes(),
RelationInstanceBean.getInstanceId(), RelationInstanceBean.getType() und
RelationInstanceBean.setInstance().

String metarelationId [private]

Definiert in Zeile 69 der Datei RelationInstanceBean.java.

Wird benutzt von RelationInstanceBean.getAttributes(),
RelationInstanceBean.getMetarelationId(), RelationInstanceBean.getType() und
RelationInstanceBean.setInstance().

String password =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")

Definiert in Zeile 54 der Datei RelationInstanceBean.java.

Wird benutzt von RelationInstanceBean.editAttribute(),
RelationInstanceBean.findDataObject(), RelationInstanceBean.getFirstMember() und
RelationInstanceBean.getSecondMember().

EAMRelation r [private]

Definiert in Zeile 77 der Datei RelationInstanceBean.java.

Wird benutzt von `RelationInstanceBean.getFirstKard()`,
`RelationInstanceBean.getFirstMember()`, `RelationInstanceBean.getFirstmembertyp()`,
`RelationInstanceBean.getSecondKard()`, `RelationInstanceBean.getSecondMember()`,
`RelationInstanceBean.getSecondmembertyp()`, `RelationInstanceBean.getType()` und
`RelationInstanceBean.setInstance()`.

String `userId =`
(String)`FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")`

Definiert in Zeile 53 der Datei `RelationInstanceBean.java`.

Wird benutzt von `RelationInstanceBean.editAttribute()`.

5.3.4 RelationInstancesBean Klassenreferenz

5.3.4.1 Ausführliche Beschreibung

Autor:

Christian Zillmann Diese Klasse dient zum Auflisten von Instanzen von Metarelationen.

Definiert in Zeile 53 der Datei `RelationInstancesBean.java`.

Öffentliche Methoden

- `String addMetamodelFilter ()`
- `String addMetarelationFilter ()`
- `String cancel ()`
- `String changeMetaRelationFunction ()`
- `String deleteInstance ()`
- `javax.faces.component.html.HtmlCommandButton getChangeMetaRelation ()`
- `javax.faces.component.html.HtmlCommandButton getCreateRelationButton ()`
- `javax.faces.component.html.HtmlSelectOneMenu getFirstMember ()`
- `List< SelectItem > getFirstMemberMenu ()`
- `String getMetamodelFilter ()`
- `String getMetamodelFilterFromSession ()`
- `Long getMetamodelFilterIdFromSession ()`
- `List< SelectItem > getMetaModelsForSelectOne ()`
- `javax.faces.component.html.HtmlSelectOneMenu getMetaRelation ()`
- `String getMetarelationFilter ()`
- `String getMetarelationFilterFromSession ()`
- `List< SelectItem > getMetaRelationsForSelectOne ()`

- String getNewCategoryId ()
- String getNewName ()
- ArrayList< RelationBean > getRelationInstances ()
- String getSearchField ()
- String getSearchFilterFromSession ()
- javax.faces.component.html.HtmlSelectOneMenu getSecondMember ()
- List< SelectItem > getSecondMemberMenu ()
- javax.faces.component.html.HtmlCommandButton getSelectMetaRelation ()
- boolean isSetNewInstanceRelation ()
- String makeInstance ()
- RelationInstancesBean ()
- String removeMetamodelFilter ()
- String removeMetarelationFilter ()
- void removeSearchFilter ()
- void search ()
- String selectMetaRelationFunction ()
- void setChangeMetaRelation (javax.faces.component.html.HtmlCommandButton changeMetaRelation)
- void setCreateRelationButton (javax.faces.component.html.HtmlCommandButton createRelationButton)
- void setFirstMember (javax.faces.component.html.HtmlSelectOneMenu firstMember)
- void setMetamodelFilter (String metamodelFilter)
- void setMetaRelation (javax.faces.component.html.HtmlSelectOneMenu metaRelation)
- void setMetarelationFilter (String metarelationFilter)
- void setNewCategoryId (String newCategoryId)
- void setNewName (String newName)
- void setSearchField (String searchField)
- void setSecondMember (javax.faces.component.html.HtmlSelectOneMenu secondMember)
- void setSelectMetaRelation (javax.faces.component.html.HtmlCommandButton selectMetaRelation)

Öffentliche Attribute

- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Private Methoden

- `DataObject findDataObject (String meta, String id)`

Private Attribute

- `javax.faces.component.html.HtmlCommandButton changeMetaRelation`
- `javax.faces.component.html.HtmlCommandButton createRelationButton`
- `EAMCategoryDAO eamCategoryDao`
- `EAMRelationDAO eamRelationDao`
- `javax.faces.component.html.HtmlSelectOneMenu firstMember`
- `String metamodelFilter`
- `javax.faces.component.html.HtmlSelectOneMenu metaRelation`
- `String metarelationFilter`
- `String newCategoryId`
- `String newName`
- `String searchField`
- `javax.faces.component.html.HtmlSelectOneMenu secondMember`
- `javax.faces.component.html.HtmlCommandButton selectMetaRelation`

5.3.4.2 Beschreibung der Konstruktoren und Destruktoren

RelationInstancesBean ()

Definiert in Zeile 95 der Datei `RelationInstancesBean.java`.

5.3.4.3 Dokumentation der Elementfunktionen

String addMetamodelFilter ()

Fügt einen MetamodellFilter hinzu

Rückgabe:

JSF-Workflow

Definiert in Zeile 290 der Datei `RelationInstancesBean.java`.

Benutzt `RelationInstancesBean.metamodelFilter`.

Wird benutzt von `RelationInstancesBean.getMetamodelFilterFromSession()` und `RelationInstancesBean.getRelationInstances()`.

String addMetarelationFilter ()

Fügt einen MetarelationsFilter hinzu.

Rückgabe:

JSF-Workflow

Definiert in Zeile 306 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.metarelationFilter.

String cancel ()

Definiert in Zeile 997 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.changeMetaRelation, RelationInstancesBean.createRelationButton, RelationInstancesBean.firstMember, RelationInstancesBean.metaRelation, RelationInstancesBean.secondMember und RelationInstancesBean.selectMetaRelation.

String changeMetaRelationFunction ()

Definiert in Zeile 949 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.changeMetaRelation, RelationInstancesBean.createRelationButton, RelationInstancesBean.firstMember, RelationInstancesBean.metaRelation, RelationInstancesBean.secondMember und RelationInstancesBean.selectMetaRelation.

String deleteInstance ()

Löscht eine Instanz einer Metarelation

Rückgabe:

JSF-Workflow

Definiert in Zeile 675 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.findDataObject() und RelationInstancesBean.password.

DataObject findDataObject (String meta, String id) [private]

Findet ein DataObject mittels der ID einer Metarelation und der ID einer Instanz einer Metarelation.

Parameter:

meta ID der Metarelation

id ID der Instanz

Rückgabe:

DataObject

Definiert in Zeile 726 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.eamRelationDao und RelationInstancesBean.password.

Wird benutzt von `RelationInstancesBean.deleteInstance()`.

`javax.faces.component.html.HtmlCommandButton` `getChangeMetaRelation ()`

Definiert in Zeile 86 der Datei `RelationInstancesBean.java`.

Benutzt `RelationInstancesBean.changeMetaRelation`.

`javax.faces.component.html.HtmlCommandButton` `getCreateRelationButton ()`

Definiert in Zeile 978 der Datei `RelationInstancesBean.java`.

Benutzt `RelationInstancesBean.createRelationButton`.

`javax.faces.component.html.HtmlSelectOneMenu` `getFirstMember ()`

Definiert in Zeile 780 der Datei `RelationInstancesBean.java`.

Benutzt `RelationInstancesBean.firstMember`.

`List<SelectedItem>` `getFirstMemberMenu ()`

Liefert eine Liste von `SelectItems` zurück mit den Instanzen, die als `FirstMember` an der gesetzten Relation hängen.

Rückgabe:

`List<SelectedItem>`

Definiert in Zeile 803 der Datei `RelationInstancesBean.java`.

Benutzt `RelationInstancesBean.eamRelationDao`, `RelationInstancesBean.newCategoryId` und `RelationInstancesBean.password`.

`String` `getMetamodelFilter ()`

Definiert in Zeile 355 der Datei `RelationInstancesBean.java`.

Benutzt `RelationInstancesBean.metamodelFilter`.

`String` `getMetamodelFilterFromSession ()`

Liest den Namen des `MetamodelFilters` aus der Session aus.

Rückgabe:

Namen vom `MetamodelFilter` aus der Session

Definiert in Zeile 470 der Datei `RelationInstancesBean.java`.

Benutzt `RelationInstancesBean.addMetamodelFilter()`,
`RelationInstancesBean.eamCategoryId`,

RelationInstancesBean.getMetamodelFilterIdFromSession() und
RelationInstancesBean.metamodelFilter.

Long getMetamodelFilterIdFromSession ()

Liest die ID des MetamodelFilters aus der Session aus.

Rückgabe:

Id vom MetamodelFilter aus der Session

Definiert in Zeile 496 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.eamCategoryDao und
RelationInstancesBean.metamodelFilter.

Wird benutzt von RelationInstancesBean.getMetamodelFilterFromSession() und
RelationInstancesBean.getRelationInstances().

List<SelectItem> getMetaModelsForSelectOne ()

Gibt eine Liste von SelectItems mit den Metamodellen zurück

Rückgabe:

Liste von SelectItems mit vorhanden Metamodellen

Definiert in Zeile 454 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.eamCategoryDao.

javax.faces.component.html.HtmlSelectOneMenu getMetaRelation ()

Definiert in Zeile 931 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.metaRelation.

String getMetarelationFilter ()

Definiert in Zeile 359 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.metarelationFilter.

String getMetarelationFilterFromSession ()

Liest den MetarelationsFilter aus, der in der Session abgelegt wurde

Rückgabe:

MetarelationsFilter aus der Session mit Namen

Definiert in Zeile 341 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.eamRelationDao und
RelationInstancesBean.metarelationFilter.

List<SelectItem> getMetaRelationsForSelectOne ()

Listet alle Metarelationen auf und gibt eine Liste von SelectItems für eine SelectOneBox zurück

Rückgabe:

Liste von SelectItems

Definiert in Zeile 418 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.eamCategoryDao, RelationInstancesBean.eamRelationDao und RelationInstancesBean.metamodelFilter.

String getNewCategoryId ()

Definiert in Zeile 521 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.newCategoryId.

String getNewName ()

Definiert in Zeile 513 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.newName.

ArrayList<RelationBean> getRelationInstances ()

Liefert alle Instanzen von Relationen.

Rückgabe:

ArrayList<RelationBean>

Definiert in Zeile 102 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.addMetamodelFilter(),
RelationInstancesBean.eamCategoryDao, RelationInstancesBean.eamRelationDao,
RelationBean.getFirstMember(),
RelationInstancesBean.getMetamodelFilterIdFromSession(),
RelationBean.getSecondMember(), RelationBean.getType(),
RelationInstancesBean.metamodelFilter, RelationInstancesBean.metarelationFilter,
RelationInstancesBean.password, InstanceBean.setCategory(),
RelationBean.setCategoryId(), RelationBean.setCategoryName(),
InstanceBean.setEamobject(), InstanceBean.setId(), RelationBean.setId(),
RelationBean.setKardone(), RelationBean.setKardttwo(), InstanceBean.setLabel(),
RelationBean.setLabel(), RelationBean.setLock(), RelationBean.setMetaId(),
RelationBean.setType(), RelationBean.setType_first_member(),
RelationBean.setType_first_member_id(), RelationBean.setType_second_member() und
RelationBean.setType_second_member_id().

String getSearchField ()

Definiert in Zeile 397 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.searchField.

String getSearchFilterFromSession ()

Liest den SuchFilter aus, der in der Session abgelegt wurde

Rückgabe:

SuchFilter aus der Session mit Suchwort

Definiert in Zeile 366 der Datei RelationInstancesBean.java.

javax.faces.component.html.HtmlSelectOneMenu getSecondMember ()

Definiert in Zeile 789 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.secondMember.

List<SelectItem> getSecondMemberMenu ()

Liefert eine Liste von SelectItems zurück mit den Instanzen, die als SecondMember an der gesetzten Relation hängen.

Rückgabe:

List<SelectItem>

Definiert in Zeile 870 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.eamRelationDao, RelationInstancesBean.newCategoryId und RelationInstancesBean.password.

javax.faces.component.html.HtmlCommandButton getSelectMetaRelation ()

Definiert in Zeile 940 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.selectMetaRelation.

boolean isSetNewInstanceRelation ()

Definiert in Zeile 987 der Datei RelationInstancesBean.java.

String makeInstance ()

Erzeugt eine neue Instanz einer Metarelation.

Rückgabe:

JSF-Workflow

Definiert in Zeile 533 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.changeMetaRelation, RelationInstancesBean.createRelationButton, RelationInstancesBean.eamRelationDao, RelationInstancesBean.firstMember, RelationInstancesBean.metaRelation, RelationInstancesBean.newCategoryId, RelationInstancesBean.newName, RelationInstancesBean.password, RelationInstancesBean.secondMember und RelationInstancesBean.selectMetaRelation.

String removeMetamodelFilter ()

Entfernt einen MetamodellFilter

Rückgabe:

JSF-Workflow

Definiert in Zeile 318 der Datei RelationInstancesBean.java.

String removeMetarelationFilter ()

Entfernt einen MetarelationFilter

Rückgabe:

JSF-Workflow

Definiert in Zeile 329 der Datei RelationInstancesBean.java.

void removeSearchFilter ()

Entfernt den SuchFilter wieder

Definiert in Zeile 379 der Datei RelationInstancesBean.java.

void search ()

Führt eine Suche durch und setzt den SuchFilter

Definiert in Zeile 389 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.searchField.

String selectMetaRelationFunction ()

Definiert in Zeile 963 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.changeMetaRelation, RelationInstancesBean.createRelationButton, RelationInstancesBean.firstMember, RelationInstancesBean.metaRelation, RelationInstancesBean.newCategoryId, RelationInstancesBean.secondMember und RelationInstancesBean.selectMetaRelation.

void setChangeMetaRelation (javax.faces.component.html.HtmlCommandButton changeMetaRelation)

Definiert in Zeile 90 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.changeMetaRelation.

void setCreateRelationButton (javax.faces.component.html.HtmlCommandButton createRelationButton)

Definiert in Zeile 982 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.createRelationButton.

void setFirstMember (javax.faces.component.html.HtmlSelectOneMenu firstMember)

Definiert in Zeile 784 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.firstMember.

void setMetamodelFilter (String metamodelFilter)

Definiert in Zeile 405 der Datei RelationInstancesBean.java.

void setMetaRelation (javax.faces.component.html.HtmlSelectOneMenu metaRelation)

Definiert in Zeile 935 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.metaRelation.

void setMetarelationFilter (String metarelationFilter)

Definiert in Zeile 409 der Datei RelationInstancesBean.java.

void setNewCategoryId (String newCategoryId)

Definiert in Zeile 525 der Datei RelationInstancesBean.java.

void setNewName (String newName)

Definiert in Zeile 517 der Datei RelationInstancesBean.java.

void setSearchField (String searchField)

Definiert in Zeile 401 der Datei RelationInstancesBean.java.

void setSecondMember (javax.faces.component.html.HtmlSelectOneMenu secondMember)

Definiert in Zeile 793 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.secondMember.

void setSelectMetaRelation (javax.faces.component.html.HtmlCommandButton selectMetaRelation)

Definiert in Zeile 944 der Datei RelationInstancesBean.java.

Benutzt RelationInstancesBean.selectMetaRelation.

5.3.4.4 Dokumentation der Datenelemente

javax.faces.component.html.HtmlCommandButton changeMetaRelation [private]

Definiert in Zeile 80 der Datei RelationInstancesBean.java.

Wird benutzt von RelationInstancesBean.chancel(),
RelationInstancesBean.changeMetaRelationFunction(),
RelationInstancesBean.getChangeMetaRelation(), RelationInstancesBean.makeInstance(),
RelationInstancesBean.selectMetaRelationFunction() und
RelationInstancesBean.setChangeMetaRelation().

javax.faces.component.html.HtmlCommandButton createRelationButton [private]

Definiert in Zeile 84 der Datei RelationInstancesBean.java.

Wird benutzt von RelationInstancesBean.chancel(),
RelationInstancesBean.changeMetaRelationFunction(),
RelationInstancesBean.createRelationButton(),
RelationInstancesBean.makeInstance(),
RelationInstancesBean.selectMetaRelationFunction() und
RelationInstancesBean.setCreateRelationButton().

EAMCategoryDAO eamCategoryDao [private]

Initialisierung:

```
DAOFactory.getAuthInstance(userId, password)
            .getEAMCategoryDAO()
```

Definiert in Zeile 61 der Datei RelationInstancesBean.java.

Wird benutzt von RelationInstancesBean.getMetamodelFilterFromSession(),
RelationInstancesBean.getMetamodelFilterIdFromSession(),
RelationInstancesBean.getMetaModelsForSelectOne(),
RelationInstancesBean.getMetaRelationsForSelectOne() und
RelationInstancesBean.getRelationInstances().

EAMRelationDAO eamRelationDao [private]

Initialisierung:

```
DAOFactory.getAuthInstance(userId, password)
            .getEAMRelationDAO()
```

Definiert in Zeile 58 der Datei RelationInstancesBean.java.

Wird benutzt von RelationInstancesBean.findDataObject(),
RelationInstancesBean.getFirstMemberMenu(),
RelationInstancesBean.getMetarelationFilterFromSession(),
RelationInstancesBean.getMetaRelationsForSelectOne(),
RelationInstancesBean.getRelationInstances(),
RelationInstancesBean.getSecondMemberMenu() und
RelationInstancesBean.makeInstance().

javax.faces.component.html.HtmlSelectOneMenu firstMember [private]

Definiert in Zeile 74 der Datei RelationInstancesBean.java.

Wird benutzt von RelationInstancesBean.chancel(),
RelationInstancesBean.changeMetaRelationFunction(),
RelationInstancesBean.getFirstMember(), RelationInstancesBean.makeInstance(),
RelationInstancesBean.selectMetaRelationFunction() und
RelationInstancesBean.setFirstMember().

String metamodelFilter [private]

Definiert in Zeile 68 der Datei RelationInstancesBean.java.

Wird benutzt von RelationInstancesBean.addMetamodelFilter(),
RelationInstancesBean.getMetamodelFilter(),
RelationInstancesBean.getMetamodelFilterFromSession(),
RelationInstancesBean.getMetamodelFilterIdFromSession(),
RelationInstancesBean.getMetaRelationsForSelectOne() und
RelationInstancesBean.getRelationInstances().

javax.faces.component.html.HtmlSelectOneMenu metaRelation [private]

Definiert in Zeile 78 der Datei RelationInstancesBean.java.

Wird benutzt von RelationInstancesBean.chancel(),
RelationInstancesBean.changeMetaRelationFunction(),
RelationInstancesBean.getMetaRelation(), RelationInstancesBean.makeInstance(),
RelationInstancesBean.selectMetaRelationFunction() und
RelationInstancesBean.setMetaRelation().

String metarelationFilter [private]

Definiert in Zeile 70 der Datei RelationInstancesBean.java.

Wird benutzt von RelationInstancesBean.addMetarelationFilter(),
RelationInstancesBean.getMetarelationFilter(),
RelationInstancesBean.getMetarelationFilterFromSession() und
RelationInstancesBean.getRelationInstances().

String newCategoryId [private]

Definiert in Zeile 66 der Datei RelationInstancesBean.java.

Wird benutzt von RelationInstancesBean.getFirstMemberMenu(),
RelationInstancesBean.getNewCategoryId(),
RelationInstancesBean.getSecondMemberMenu(), RelationInstancesBean.makeInstance()
und RelationInstancesBean.selectMetaRelationFunction().

String newName [private]

Definiert in Zeile 64 der Datei RelationInstancesBean.java.

Wird benutzt von RelationInstancesBean.getNewName() und
RelationInstancesBean.makeInstance().

String password =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")

Definiert in Zeile 56 der Datei RelationInstancesBean.java.

Wird benutzt von RelationInstancesBean.deleteInstance(),
RelationInstancesBean.findDataObject(), RelationInstancesBean.getFirstMemberMenu(),
RelationInstancesBean.getRelationInstances(),
RelationInstancesBean.getSecondMemberMenu() und
RelationInstancesBean.makeInstance().

String searchField [private]

Definiert in Zeile 72 der Datei RelationInstancesBean.java.

Wird benutzt von RelationInstancesBean.getSearchField() und
RelationInstancesBean.search().

javax.faces.component.html.HtmlSelectOneMenu secondMember [private]

Definiert in Zeile 76 der Datei RelationInstancesBean.java.

Wird benutzt von RelationInstancesBean.chancel(),
RelationInstancesBean.changeMetaRelationFunction(),

RelationInstancesBean.getSecondMember(), RelationInstancesBean.makeInstance(),
RelationInstancesBean.selectMetaRelationFunction() und
RelationInstancesBean.setSecondMember().

javax.faces.component.html.HtmlCommandButton selectMetaRelation [private]

Definiert in Zeile 82 der Datei RelationInstancesBean.java.

Wird benutzt von RelationInstancesBean.chancel(),
RelationInstancesBean.changeMetaRelationFunction(),
RelationInstancesBean.getSelectMetaRelation(), RelationInstancesBean.makeInstance(),
RelationInstancesBean.selectMetaRelationFunction() und
RelationInstancesBean.setSelectMetaRelation().

String **userId =**
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Definiert in Zeile 55 der Datei RelationInstancesBean.java.

5.4 Tag-Library

5.4.1 AttributesTag Klassenreferenz

Abgeleitet von javax::servlet::jsp::tagext::TagSupport.

5.4.1.1 Ausführliche Beschreibung

Autor:

Christian Zillmann Dies ist eine Tag-Klasse. Die Tags dieser Klasse listen dynamisch
soviele Feld zur Eingabe auf, wie es Attribute gibt.

Definiert in Zeile 28 der Datei AttributesTag.java.

Öffentliche Methoden

- int doAfterBody () throws JspException
- int doEndTag () throws JspException
- int doStartTag () throws JspException
- void setPageContext (PageContext pageContext)
- void setType (String type)

Öffentliche Attribute

- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Private Methoden

- ArrayList< String > parseEnum (String enumtype)

Private Attribute

- String type

Statische private Attribute

- static final long serialVersionUID = 1L

5.4.1.2 Dokumentation der Elementfunktionen

int doAfterBody () throws JspException

Definiert in Zeile 40 der Datei AttributesTag.java.

int doEndTag () throws JspException

Definiert in Zeile 45 der Datei AttributesTag.java.

int doStartTag () throws JspException

Definiert in Zeile 50 der Datei AttributesTag.java.

Benutzt AttributesTag.parseEnum(), AttributesTag.password, AttributesTag.type und AttributesTag.userId.

ArrayList<String> parseEnum (String enumtype) [private]

Definiert in Zeile 204 der Datei AttributesTag.java.

Wird benutzt von AttributesTag.doStartTag().

void setPageContext (PageContext pageContext)

Definiert in Zeile 196 der Datei AttributesTag.java.

void setType (String type)

Definiert in Zeile 200 der Datei AttributesTag.java.

5.4.1.3 Dokumentation der Datenelemente

```
String password =  
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
```

Definiert in Zeile 31 der Datei AttributesTag.java.

Wird benutzt von AttributesTag.doStartTag().

```
final long serialVersionUID = 1L [static, private]
```

Definiert in Zeile 37 der Datei AttributesTag.java.

```
String type [private]
```

Definiert in Zeile 33 der Datei AttributesTag.java.

Wird benutzt von AttributesTag.doStartTag().

```
String userId =  
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")
```

Definiert in Zeile 30 der Datei AttributesTag.java.

Wird benutzt von AttributesTag.doStartTag().

5.4.2 EditAttributeTag Klassenreferenz

Abgeleitet von javax:servlet:jsp::tagext::TagSupport.

5.4.2.1 Ausführliche Beschreibung**Autor:**

Christian Zillmann Dies ist eine Tag-Klasse. Die Tags dieser Klasse können dazu verwendet werden, kleine Fenster einzublenden wenn ein Eingabefeld eines Attributes den Fokus bekommt. Diese Fenster geben dann Informationen zu dem Attribut preis.

Definiert in Zeile 34 der Datei EditAttributeTag.java.

Öffentliche Methoden

- int doAfterBody () throws JspException

- `int doEndTag ()` throws `JspException`
- `int doStartTag ()` throws `JspException`
- `String getAttributeId ()`
- `String getInstanceId ()`
- `String getMetaId ()`
- `String getObjectOrRelation ()`
- `void setAttributeId (String attributeId)`
- `void setInstanceId (String instanceId)`
- `void setMetaId (String metaId)`
- `void setObjectOrRelation (String objectOrRelation)`
- `void setPageContext (PageContext pageContext)`

Öffentliche Attribute

- `String password =`
`(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")`
- `String userId =`
`(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")`

Private Methoden

- `ArrayList< String > parseEnum (String enumtype)`

Private Attribute

- `String attributeId`
- `String instanceId`
- `String metaId`
- `String objectOrRelation`

5.4.2.2 Dokumentation der Elementfunktionen

`int doAfterBody ()` throws `JspException`

Definiert in Zeile 89 der Datei `EditAttributeTag.java`.

`int doEndTag ()` throws `JspException`

Definiert in Zeile 100 der Datei `EditAttributeTag.java`.

`int doStartTag ()` throws `JspException`

Definiert in Zeile 109 der Datei EditAttributeTag.java.

Benutzt EditAttributeTag.attributeId, EditAttributeTag.instanceId, EditAttributeTag.metaId, EditAttributeTag.objectOrRelation, EditAttributeTag.parseEnum(), EditAttributeTag.password und EditAttributeTag.userId.

String getAttributeld ()

Definiert in Zeile 75 der Datei EditAttributeTag.java.

Benutzt EditAttributeTag.attributeId.

String getInstanceId ()

Definiert in Zeile 51 der Datei EditAttributeTag.java.

Benutzt EditAttributeTag.instanceId.

String getMetaId ()

Definiert in Zeile 67 der Datei EditAttributeTag.java.

Benutzt EditAttributeTag.metaId.

String getObjectOrRelation ()

Definiert in Zeile 59 der Datei EditAttributeTag.java.

Benutzt EditAttributeTag.objectOrRelation.

ArrayList<String> parseEnum (String enumtype) [private]

Definiert in Zeile 237 der Datei EditAttributeTag.java.

Wird benutzt von EditAttributeTag.doStartTag().

void setAttributeld (String attributeld)

Definiert in Zeile 79 der Datei EditAttributeTag.java.

void setInstanceId (String instanceId)

Definiert in Zeile 55 der Datei EditAttributeTag.java.

void setMetaId (String metaId)

Definiert in Zeile 71 der Datei EditAttributeTag.java.

void setObjectOrRelation (String objectOrRelation)

Definiert in Zeile 63 der Datei EditAttributeTag.java.

void setPageContext (PageContext pageContext)

Definiert in Zeile 232 der Datei EditAttributeTag.java.

5.4.2.3 Dokumentation der Datenelemente

String attributeId [private]

Definiert in Zeile 47 der Datei EditAttributeTag.java.

Wird benutzt von EditAttributeTag.doStartTag() und EditAttributeTag.getAttributeId().

String instanceId [private]

Definiert in Zeile 49 der Datei EditAttributeTag.java.

Wird benutzt von EditAttributeTag.doStartTag() und EditAttributeTag.getInstanceId().

String metaId [private]

Definiert in Zeile 45 der Datei EditAttributeTag.java.

Wird benutzt von EditAttributeTag.doStartTag() und EditAttributeTag.getMetaId().

String objectOrRelation [private]

Definiert in Zeile 43 der Datei EditAttributeTag.java.

Wird benutzt von EditAttributeTag.doStartTag() und EditAttributeTag.getObjectOrRelation().

String password =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")

Definiert in Zeile 39 der Datei EditAttributeTag.java.

Wird benutzt von EditAttributeTag.doStartTag().

String userId =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Definiert in Zeile 38 der Datei EditAttributeTag.java.

Wird benutzt von EditAttributeTag.doStartTag().

5.4.3 MessageTag Klassenreferenz

Abgeleitet von javax::servlet::jsp::tagext::TagSupport.

5.4.3.1 Ausführliche Beschreibung

Autor:

Christian Zillmann Diese Klasse ist fuer die Taglibrary des Sicherheitssystem. Mit Hilfe dieses Tags werden Bereiche auf einer Seite mit einer Rolle/ID/Recht gekennzeichnet. Diese/s Rolle/ID/Recht wird spaeter von Admin dann auf Rollen gemappt die existieren.

christian zillmann

Definiert in Zeile 23 der Datei MessageTag.java.

Öffentliche Methoden

- int doAfterBody () throws JspException
- int doEndTag () throws JspException
- int doStartTag () throws JspException
- void setMessageAttribut (String messageAttribut)
- void setPageContext (PageContext pageContext)

Private Attribute

- String messageAttribut

Statische private Attribute

- static final long serialVersionUID = 1L

5.4.3.2 Dokumentation der Elementfunktionen

int doAfterBody () throws JspException

Definiert in Zeile 44 der Datei MessageTag.java.

int doEndTag () throws JspException

Definiert in Zeile 55 der Datei MessageTag.java.

int doStartTag () throws JspException

Definiert in Zeile 64 der Datei MessageTag.java.

Benutzt MessageTag.messageAttribut.

void setMessageAttribut (String messageAttribut)

Definiert in Zeile 34 der Datei MessageTag.java.

void setPageContext (PageContext pageContext)

Definiert in Zeile 93 der Datei MessageTag.java.

5.4.3.3 Dokumentation der Datenelemente

String messageAttribut [private]

Definiert in Zeile 30 der Datei MessageTag.java.

Wird benutzt von MessageTag.doStartTag().

final long serialVersionUID = 1L [static, private]

Definiert in Zeile 28 der Datei MessageTag.java.

5.5 Sonstiges

5.5.1 Tools Klassenreferenz

5.5.1.1 Ausführliche Beschreibung

Definiert in Zeile 3 der Datei Tools.java.

6 de.offis.pg.eam.mod_querybrowser

6.0.2 BundleViews Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::module::api::IModuleView.

6.0.2.1 Ausführliche Beschreibung

Definiert in Zeile 22 der Datei BundleViews.java.

Öffentliche Methoden

- String getBundleVersion ()
- String getDescription ()
- long getEquinoxBundleId ()
- String getName ()
- List< View > getViews ()

Statische private Attribute

- static final String VIEWS = "bundle_views.xml"

6.0.2.2 Dokumentation der Elementfunktionen

String getBundleVersion ()

Definiert in Zeile 68 der Datei BundleViews.java.

String getDescription ()

Definiert in Zeile 29 der Datei BundleViews.java.

long getEquinoxBundleId ()

Definiert in Zeile 34 der Datei BundleViews.java.

String getName ()

Definiert in Zeile 39 der Datei BundleViews.java.

List<View> getViews ()

Definiert in Zeile 44 der Datei BundleViews.java.

Benutzt BundleViews.VIEWS.

6.0.2.3 Dokumentation der Datenelemente

final String VIEWS = "bundle_views.xml" [static, private]

Definiert in Zeile 25 der Datei BundleViews.java.

Wird benutzt von BundleViews.getViews().

6.0.3 Configuration Klassenreferenz

Abgeleitet von *de::offis::pg::eam::core::configuration::api::IBundleConfiguration*.

6.0.3.1 Ausführliche Beschreibung

Definiert in Zeile 10 der Datei Configuration.java.

Öffentliche Methoden

- boolean bundleHasObjects ()
- boolean bundleHasRelations ()
- boolean bundleHasViews ()
- Document getObjectsJdomDocument () throws JDOMException, IOException

6.0.3.2 Dokumentation der Elementfunktionen

boolean bundleHasObjects ()

Definiert in Zeile 13 der Datei Configuration.java.

boolean bundleHasRelations ()

Definiert in Zeile 28 der Datei Configuration.java.

boolean bundleHasViews ()

Definiert in Zeile 18 der Datei Configuration.java.

Document `getObjectsJdomDocument ()` throws **JDOMException**, **IOException**

Definiert in Zeile 23 der Datei `Configuration.java`.

6.0.4 Activator Klassenreferenz

Abgeleitet von `org::osgi::framework::BundleActivator`.

6.0.4.1 Ausführliche Beschreibung

Definiert in Zeile 33 der Datei `Activator.java`.

Öffentliche Methoden

- `void start (BundleContext context) throws Exception`
- `void stop (BundleContext context) throws Exception`

Statische öffentliche Attribute

- `static BundleContext bundleContext`
- `static String bundleSymbolicName`
- `static String bundleVersion`
- `static long equinoxBundleId`
- `static final String PATH = "/querybrowser"`

Private Attribute

- `ServiceTracker httpServiceTracker`

Statische private Attribute

- `static final String WEB_ROOT = "/WebRoot"`

Klassen

- `class HttpServiceTracker`
- `class ServletContextListenerServletAdaptor`

6.0.4.2 Dokumentation der Elementfunktionen

`void start (BundleContext context) throws Exception`

Definiert in Zeile 47 der Datei Activator.java.

Benutzt Activator.bundleContext, Activator.bundleSymbolicName, Activator.bundleVersion, Activator.equinoxBundleId und Activator.httpServiceTracker.

void stop (BundleContext context) throws Exception

Definiert in Zeile 60 der Datei Activator.java.

Benutzt Activator.httpServiceTracker.

6.0.4.3 Dokumentation der Datenelemente

BundleContext bundleContext [static]

Definiert in Zeile 42 der Datei Activator.java.

Wird benutzt von Activator.start().

String bundleSymbolicName [static]

Definiert in Zeile 41 der Datei Activator.java.

Wird benutzt von Activator.start().

String bundleVersion [static]

Definiert in Zeile 43 der Datei Activator.java.

Wird benutzt von Activator.start().

long equinoxBundleId [static]

Definiert in Zeile 40 der Datei Activator.java.

Wird benutzt von Activator.start().

ServiceTracker httpServiceTracker [private]

Definiert in Zeile 37 der Datei Activator.java.

Wird benutzt von Activator.start() und Activator.stop().

final String PATH = "/querybrowser" [static]

Definiert in Zeile 35 der Datei Activator.java.

Wird benutzt von Activator.HttpServiceTracker.addingService() und Activator.HttpServiceTracker.removedService().

final String WEB_ROOT = "/WebRoot" [static, private]

Definiert in Zeile 36 der Datei Activator.java.

Wird benutzt von Activator.HttpServiceTracker.addingService().

6.0.5 Activator.HttpServiceTracker Klassenreferenz

Abgeleitet von org::osgi::util::tracker::ServiceTracker.

6.0.5.1 Ausführliche Beschreibung

Definiert in Zeile 64 der Datei Activator.java.

Öffentliche Methoden

- Object addingService (ServiceReference reference)
- HttpServiceTracker (BundleContext context)
- void removedService (ServiceReference reference, Object service)

6.0.5.2 Beschreibung der Konstruktoren und Destruktoren

HttpServiceTracker (BundleContext context)

Definiert in Zeile 66 der Datei Activator.java.

6.0.5.3 Dokumentation der Elementfunktionen

Object addingService (ServiceReference reference)

Definiert in Zeile 70 der Datei Activator.java.

Benutzt Activator.PATH und Activator.WEB_ROOT.

void removedService (ServiceReference reference, Object service)

Definiert in Zeile 140 der Datei Activator.java.

Benutzt Activator.PATH.

6.0.6 Activator.ServletContextListenerServletAdaptor Klassenreferenz

Abgeleitet von javax::servlet::Servlet.

6.0.6.1 Ausführliche Beschreibung

Definiert in Zeile 156 der Datei Activator.java.

Öffentliche Methoden

- void destroy ()
- ServletConfig getServletConfig ()
- String getServletInfo ()
- void init (ServletConfig config) throws ServletException
- void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException
- ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Private Attribute

- ServletConfig config
- Servlet delegate
- ClassLoader jspLoader
- ServletContextListener listener

6.0.6.2 Beschreibung der Konstruktoren und Destruktoren

ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Definiert in Zeile 166 der Datei Activator.java.

6.0.6.3 Dokumentation der Elementfunktionen

void destroy ()

Definiert in Zeile 228 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config, Activator.ServletContextListenerServletAdaptor.delegate, Activator.ServletContextListenerServletAdaptor.jspLoader und Activator.ServletContextListenerServletAdaptor.listener.

ServletConfig getServletConfig ()

Definiert in Zeile 254 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config.

String getServletInfo ()

Definiert in Zeile 260 der Datei Activator.java.

void init (ServletConfig config) throws ServletException

Definiert in Zeile 180 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config, Activator.ServletContextListenerServletAdaptor.delegate, Activator.ServletContextListenerServletAdaptor.jspLoader und Activator.ServletContextListenerServletAdaptor.listener.

void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException

Definiert in Zeile 206 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.delegate und Activator.ServletContextListenerServletAdaptor.jspLoader.

6.0.6.4 Dokumentation der Datenelemente

ServletConfig config [private]

Definiert in Zeile 158 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(), Activator.ServletContextListenerServletAdaptor.getServletConfig() und Activator.ServletContextListenerServletAdaptor.init().

Servlet delegate [private]

Definiert in Zeile 162 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(), Activator.ServletContextListenerServletAdaptor.init() und Activator.ServletContextListenerServletAdaptor.service().

ClassLoader jspLoader [private]

Definiert in Zeile 164 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(), Activator.ServletContextListenerServletAdaptor.init() und Activator.ServletContextListenerServletAdaptor.service().

ServletContextListener listener [private]

Definiert in Zeile 160 der Datei Activator.java.

Wird benutzt von `Activator.ServletContextListenerServletAdaptor.destroy()` und `Activator.ServletContextListenerServletAdaptor.init()`.

6.0.7 MenuConstants Schnittstellenreferenz

6.0.7.1 Ausführliche Beschreibung

Definiert in Zeile 3 der Datei `MenuConstants.java`.

Statische öffentliche Attribute

- `static final String ADOPTER_ID = "querybrowserAdopter"`
- `static final String CONTRIBUTOR_ID = "querybrowserContributor"`

6.0.7.2 Dokumentation der Datenelemente

`final String ADOPTER_ID = "querybrowserAdopter" [static]`

Definiert in Zeile 7 der Datei `MenuConstants.java`.

`final String CONTRIBUTOR_ID = "querybrowserContributor" [static]`

Definiert in Zeile 5 der Datei `MenuConstants.java`.

6.0.8 MenuContributor Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::menu::api::AbstractAdoptableMenuContributor`.

6.0.8.1 Ausführliche Beschreibung

Definiert in Zeile 6 der Datei `MenuContributor.java`.

Öffentliche Methoden

- `String getAdopterId ()`
- `String getContributionTarget ()`
- `String getContributorId ()`
- `String getTargetSubmenu ()`
- `MenuContributor ()`

6.0.8.2 Beschreibung der Konstruktoren und Destruktoren

MenuContributor ()

Definiert in Zeile 8 der Datei MenuContributor.java.

6.0.8.3 Dokumentation der Elementfunktionen

String getAdopterId ()

Definiert in Zeile 12 der Datei MenuContributor.java.

String getContributionTarget ()

Definiert in Zeile 16 der Datei MenuContributor.java.

String getContributorId ()

Definiert in Zeile 20 der Datei MenuContributor.java.

String getTargetSubmenu ()

Definiert in Zeile 24 der Datei MenuContributor.java.

6.1 Managed Beans

6.1.1 Query Klassenreferenz

6.1.1.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 10 der Datei Query.java.

Öffentliche Methoden

- boolean isEditable ()
- Query (String name, String description, String sql)
- Query ()
- void setEditable (boolean editable)

Private Attribute

- boolean editable = false

6.1.1.2 Beschreibung der Konstruktoren und Destruktoren

Query ()

Definiert in Zeile 15 der Datei Query.java.

Query (String name, String description, String sql)

Definiert in Zeile 18 der Datei Query.java.

6.1.1.3 Dokumentation der Elementfunktionen

boolean isEditable ()

Definiert in Zeile 22 der Datei Query.java.

Benutzt Query.editable.

void setEditable (boolean editable)

Definiert in Zeile 25 der Datei Query.java.

Wird benutzt von QueryBrowserBean.getQueries().

6.1.1.4 Dokumentation der Datenelemente

boolean editable = false [private]

Definiert in Zeile 12 der Datei Query.java.

Wird benutzt von Query.isEditable().

6.1.2 QueryBrowserBean Klassenreferenz

6.1.2.1 Ausführliche Beschreibung

Definiert in Zeile 27 der Datei QueryBrowserBean.java.

Öffentliche Methoden

- void deleteQuery ()
- String editQuery ()
- String executeQuery ()
- HtmlInputHidden getHidden ()
- LinkedList< Query > getQueries ()
- void setHidden (HtmlInputHidden hidden)

- void setMessage (String message)

Private Attribute

- HtmlInputHidden hidden
- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- HashMap< Long, Query > queries = new HashMap<Long, Query>()
- IQueryDAO queryDAO
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

6.1.2.2 Dokumentation der Elementfunktionen

void deleteQuery ()

Definiert in Zeile 210 der Datei QueryBrowserBean.java.

Benutzt QueryBrowserBean.hidden, QueryBrowserBean.password, QueryBrowserBean.queryDAO, QueryBrowserBean.setMessage() und QueryBrowserBean.userId.

String editQuery ()

Definiert in Zeile 190 der Datei QueryBrowserBean.java.

Benutzt QueryBrowserBean.hidden.

String executeQuery ()

Definiert in Zeile 119 der Datei QueryBrowserBean.java.

Benutzt QueryBrowserBean.hidden, QueryBrowserBean.password, QueryBrowserBean.queries, QueryBrowserBean.queryDAO, QueryBrowserBean.setMessage() und QueryBrowserBean.userId.

HtmlInputHidden getHidden ()

Definiert in Zeile 57 der Datei QueryBrowserBean.java.

Benutzt QueryBrowserBean.hidden.

LinkedList<Query> getQueries ()

Definiert in Zeile 67 der Datei QueryBrowserBean.java.

Benutzt QueryBrowserBean.password, QueryBrowserBean.queries, QueryBrowserBean.queryDAO, Query.setEditable(), QueryBrowserBean.setMessage() und QueryBrowserBean.userId.

void setHidden (HtmlInputHidden hidden)

Definiert in Zeile 60 der Datei QueryBrowserBean.java.

void setMessage (String message)

Definiert in Zeile 50 der Datei QueryBrowserBean.java.

Wird benutzt von QueryBrowserBean.deleteQuery(), QueryBrowserBean.executeQuery() und QueryBrowserBean.getQueries().

6.1.2.3 Dokumentation der Datenelemente

HtmlInputHidden hidden [private]

Definiert in Zeile 32 der Datei QueryBrowserBean.java.

Wird benutzt von QueryBrowserBean.deleteQuery(), QueryBrowserBean.editQuery(), QueryBrowserBean.executeQuery() und QueryBrowserBean.getHidden().

String password =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
[private]

Definiert in Zeile 30 der Datei QueryBrowserBean.java.

Wird benutzt von QueryBrowserBean.deleteQuery(), QueryBrowserBean.executeQuery() und QueryBrowserBean.getQueries().

HashMap<Long, Query> queries = new HashMap<Long, Query>() [private]

Definiert in Zeile 33 der Datei QueryBrowserBean.java.

Wird benutzt von QueryBrowserBean.executeQuery() und QueryBrowserBean.getQueries().

IQueryDAO queryDAO [private]

Initialisierung:

```
{
    try {
        queryDAO = (IQueryDAO)EAMProxy.newInstance(IQueryDAO.class, Long.valueOf(userId), password);
    } catch (NumberFormatException e) {
        e.printStackTrace();
    }
}
```



```
    } catch (ProxyException e) {  
        e.printStackTrace();  
    } catch (Throwable e) {  
  
        e.printStackTrace();  
    }  
}
```

Definiert in Zeile 35 der Datei QueryBrowserBean.java.

Wird benutzt von QueryBrowserBean.deleteQuery(), QueryBrowserBean.executeQuery() und QueryBrowserBean.getQueries().

```
String userId =  
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")  
[private]
```

Definiert in Zeile 29 der Datei QueryBrowserBean.java.

Wird benutzt von QueryBrowserBean.deleteQuery(), QueryBrowserBean.executeQuery() und QueryBrowserBean.getQueries().

6.1.3 QueryEditorBean Klassenreferenz

6.1.3.1 Ausführliche Beschreibung

Definiert in Zeile 41 der Datei QueryEditorBean.java.

Öffentliche Methoden

- void executeQuery ()
- String getAttribute ()
- LinkedList< SelectItem > getAttributes ()
- DataObjectList getDataObjectList ()
- String getFetchTime ()
- HtmlInputHidden getHidden ()
- String getMetamodel ()
- LinkedList< SelectItem > getMetamodels ()
- String getObject ()
- LinkedList< SelectItem > getObjects ()
- Query getQuery ()
- String getQueryString ()
- void newQuery ()
- void saveQuery ()

- void selectAttribute (ValueChangeEvent valueChangeEvent)
- void selectMetamodel (ValueChangeEvent valueChangeEvent)
- void selectObject (ValueChangeEvent valueChangeEvent)
- void setAttribute (String attribute)
- void setHidden (HtmlInputHidden hidden)
- void setMessage (String message)
- void setMetamodel (String metamodel)
- void setObject (String object)
- void setQuery (Query query)
- void setQueryString (String queryString)

Private Methoden

- void initAttributes ()
- void initMetamodels ()
- void initObjects ()

Private Attribute

- EAMCategoryDAO catDAO = dao.getEAMCategoryDAO()
- DAOFactory dao = DAOFactory.getAuthInstance(userId, password)
- String fetchTime = ""
- HtmlInputHidden hidden
- IInstanceQuery instanceQuery
- EAMCategory metamodel = null
- EAMObjectDAO objDAO = dao.getEAMObjectDAO()
- EAMSuperClass object = null
- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- Query query = new Query()
- IQueryDAO queryDAO
- String queryString = ""
- EAMRelationDAO relDAO = dao.getEAMRelationDAO()
- DataObjectList results = null
- LinkedList< SelectItem > selectAttributes = new LinkedList<SelectItem>()
- String selectedAttribute = null
- String selectedMetamodel = null
- String selectedObject = null
- LinkedList< SelectItem > selectMetamodels = new LinkedList<SelectItem>()

- `LinkedList< SelectItem > selectObjects = new LinkedList<SelectItem>()`
- `String userId =`
`(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")`

6.1.3.2 Dokumentation der Elementfunktionen

void executeQuery ()

Definiert in Zeile 233 der Datei `QueryEditorBean.java`.

Benutzt `QueryEditorBean.fetchTime`, `QueryEditorBean.instanceQuery`, `QueryEditorBean.password`, `QueryEditorBean.query`, `QueryEditorBean.queryString`, `QueryEditorBean.results`, `QueryEditorBean.setMessage()` und `QueryEditorBean.userId`.

String getAttribute ()

Definiert in Zeile 464 der Datei `QueryEditorBean.java`.

Benutzt `QueryEditorBean.selectedAttribute`.

LinkedList<SelectItem> getAttributes ()

Definiert in Zeile 470 der Datei `QueryEditorBean.java`.

Benutzt `QueryEditorBean.selectAttributes`.

Wird benutzt von `QueryEditorBean.initAttributes()`.

DataObjectList getDataObjectList ()

Definiert in Zeile 168 der Datei `QueryEditorBean.java`.

Benutzt `QueryEditorBean.results`.

String getFetchTime ()

Definiert in Zeile 162 der Datei `QueryEditorBean.java`.

Benutzt `QueryEditorBean.fetchTime`.

HtmlInputHidden getHidden ()

Definiert in Zeile 107 der Datei `QueryEditorBean.java`.

Benutzt `QueryEditorBean.hidden`.

String getMetamodel ()

Definiert in Zeile 331 der Datei `QueryEditorBean.java`.

Benutzt `QueryEditorBean.selectedMetamodel`.

LinkedList<SelectItem> getMetamodels ()

Definiert in Zeile 348 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.initMetamodels(), QueryEditorBean.initObjects() und QueryEditorBean.selectMetamodels.

String getObject ()

Definiert in Zeile 384 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.selectedObject.

LinkedList<SelectItem> getObjects ()

Definiert in Zeile 394 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.selectObjects.

Query getQuery ()

Definiert in Zeile 116 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.query.

String getQueryString ()

Definiert in Zeile 127 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.password, QueryEditorBean.query, QueryEditorBean.queryDAO, QueryEditorBean.queryString und QueryEditorBean.userId.

void initAttributes () [private]

Definiert in Zeile 474 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.getAttributes() und QueryEditorBean.selectAttributes.

Wird benutzt von QueryEditorBean.setObject().

void initMetamodels () [private]

Definiert in Zeile 356 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.catDAO, QueryEditorBean.metamodel und QueryEditorBean.selectMetamodels.

Wird benutzt von QueryEditorBean.getMetamodels().

void initObjects () [private]

Definiert in Zeile 397 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.metamodel, QueryEditorBean.objDAO, QueryEditorBean.relDAO und QueryEditorBean.selectObjects.

Wird benutzt von QueryEditorBean.getMetamodels() und QueryEditorBean.setMetamodel().

void newQuery ()

Definiert in Zeile 174 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.query, QueryEditorBean.queryString und QueryEditorBean.results.

void saveQuery ()

Definiert in Zeile 186 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.password, QueryEditorBean.query, QueryEditorBean.queryDAO, QueryEditorBean.setMessage() und QueryEditorBean.userId.

void selectAttribute (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 492 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.selectedAttribute.

void selectMetamodel (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 374 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.catDAO, QueryEditorBean.metamodel, QueryEditorBean.selectedMetamodel und QueryEditorBean.setMetamodel().

void selectObject (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 437 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.objDAO, QueryEditorBean.relDAO, QueryEditorBean.selectedObject und QueryEditorBean.setObject().

void setAttribute (String attribute)

Definiert in Zeile 467 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.selectedAttribute.

void setHidden (HtmlInputHidden hidden)

Definiert in Zeile 110 der Datei QueryEditorBean.java.

void setMessage (String message)

Definiert in Zeile 100 der Datei QueryEditorBean.java.

Wird benutzt von QueryEditorBean.executeQuery() und QueryEditorBean.saveQuery().

void setMetamodel (String metamodel)

Definiert in Zeile 334 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.initObjects(), QueryEditorBean.selectAttributes, QueryEditorBean.selectedAttribute, QueryEditorBean.selectedMetamodel, QueryEditorBean.selectedObject und QueryEditorBean.selectObjects.

Wird benutzt von QueryEditorBean.selectMetamodel().

void setObject (String object)

Definiert in Zeile 387 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.initAttributes() und QueryEditorBean.selectedObject.

Wird benutzt von QueryEditorBean.selectObject().

void setQuery (Query query)

Definiert in Zeile 120 der Datei QueryEditorBean.java.

void setQueryString (String queryString)

Definiert in Zeile 155 der Datei QueryEditorBean.java.

Benutzt QueryEditorBean.query.

6.1.3.3 Dokumentation der Datenelemente

EAMCategoryDAO catDAO = dao.getEAMCategoryDAO() [private]

Definiert in Zeile 47 der Datei QueryEditorBean.java.

Wird benutzt von QueryEditorBean.initMetamodels() und QueryEditorBean.selectMetamodel().

DAOFactory dao = DAOFactory.getAuthInstance(userId, password) [private]

Definiert in Zeile 46 der Datei QueryEditorBean.java.

String fetchTime = "" [private]

Definiert in Zeile 55 der Datei QueryEditorBean.java.

Wird benutzt von `QueryEditorBean.executeQuery()` und `QueryEditorBean.getFetchTime()`.

HtmlInputHidden hidden [private]

Definiert in Zeile 51 der Datei `QueryEditorBean.java`.

Wird benutzt von `QueryEditorBean.getHidden()`.

IInstanceQuery instanceQuery [private]

Initialisierung:

```
{
    try {
        instanceQuery = (IInstanceQuery)EAMProxy.newInstance(IInstanceQuery.class, Long.valueOf(userId), password);
    } catch (NumberFormatException e) {
        e.printStackTrace();
    } catch (ProxyException e) {
        e.printStackTrace();
    } catch (Throwable e) {
        e.printStackTrace();
    }
}
```

Definiert in Zeile 85 der Datei `QueryEditorBean.java`.

Wird benutzt von `QueryEditorBean.executeQuery()`.

EAMCategory metamodel = null [private]

Definiert in Zeile 57 der Datei `QueryEditorBean.java`.

Wird benutzt von `QueryEditorBean.initMetamodels()`, `QueryEditorBean.initObjects()` und `QueryEditorBean.selectMetamodel()`.

EAMObjectDAO objDAO = dao.getEAMObjectDAO() [private]

Definiert in Zeile 48 der Datei `QueryEditorBean.java`.

Wird benutzt von `QueryEditorBean.initObjects()` und `QueryEditorBean.selectObject()`.

EAMSuperClass object = null [private]

Definiert in Zeile 61 der Datei `QueryEditorBean.java`.

String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
[private]

Definiert in Zeile 44 der Datei QueryEditorBean.java.

Wird benutzt von QueryEditorBean.executeQuery(), QueryEditorBean.getQueryString() und QueryEditorBean.saveQuery().

Query query = new Query() [private]

Definiert in Zeile 53 der Datei QueryEditorBean.java.

Wird benutzt von QueryEditorBean.executeQuery(), QueryEditorBean.getQuery(), QueryEditorBean.getQueryString(), QueryEditorBean.newQuery(), QueryEditorBean.saveQuery() und QueryEditorBean.setQueryString().

IQueryDAO queryDAO [private]

Initialisierung:

```
{
    try {
        queryDAO = (IQueryDAO)EAMProxy.newInstance(IQueryDAO.class, Long.valueOf(userId), password);
    } catch (NumberFormatException e) {
        e.printStackTrace();
    } catch (ProxyException e) {
        e.printStackTrace();
    } catch (Throwable e) {
        e.printStackTrace();
    }
}
```

Definiert in Zeile 72 der Datei QueryEditorBean.java.

Wird benutzt von QueryEditorBean.getQueryString() und QueryEditorBean.saveQuery().

String queryString = "" [private]

Definiert in Zeile 54 der Datei QueryEditorBean.java.

Wird benutzt von QueryEditorBean.executeQuery(), QueryEditorBean.getQueryString() und QueryEditorBean.newQuery().

EAMRelationDAO relDAO = dao.getEAMRelationDAO() [private]

Definiert in Zeile 49 der Datei QueryEditorBean.java.

Wird benutzt von QueryEditorBean.initObjects() und QueryEditorBean.selectObject().

DataObjectList results = null [private]

Definiert in Zeile 69 der Datei QueryEditorBean.java.

Wird benutzt von `QueryEditorBean.executeQuery()`,
`QueryEditorBean.getDataObjectList()` und `QueryEditorBean.newQuery()`.

LinkedList<SelectItem> selectAttributes = new LinkedList<SelectItem>()
[private]

Definiert in Zeile 66 der Datei `QueryEditorBean.java`.

Wird benutzt von `QueryEditorBean.getAttributes()`, `QueryEditorBean.initAttributes()`
und `QueryEditorBean.setMetamodel()`.

String selectedAttribute = null [private]

Definiert in Zeile 65 der Datei `QueryEditorBean.java`.

Wird benutzt von `QueryEditorBean.getAttribute()`, `QueryEditorBean.selectAttribute()`,
`QueryEditorBean.setAttribute()` und `QueryEditorBean.setMetamodel()`.

String selectedMetamodel = null [private]

Definiert in Zeile 58 der Datei `QueryEditorBean.java`.

Wird benutzt von `QueryEditorBean.getMetamodel()`, `QueryEditorBean.selectMetamodel()`
und `QueryEditorBean.setMetamodel()`.

String selectedObject = null [private]

Definiert in Zeile 62 der Datei `QueryEditorBean.java`.

Wird benutzt von `QueryEditorBean.getObject()`, `QueryEditorBean.selectObject()`,
`QueryEditorBean.setMetamodel()` und `QueryEditorBean.setObject()`.

LinkedList<SelectItem> selectMetamodels = new LinkedList<SelectItem>()
[private]

Definiert in Zeile 59 der Datei `QueryEditorBean.java`.

Wird benutzt von `QueryEditorBean.getMetamodels()` und
`QueryEditorBean.initMetamodels()`.

LinkedList<SelectItem> selectObjects = new LinkedList<SelectItem>() [private]

Definiert in Zeile 63 der Datei `QueryEditorBean.java`.

Wird benutzt von `QueryEditorBean.getObjects()`, `QueryEditorBean.initObjects()` und
`QueryEditorBean.setMetamodel()`.

```
String userId =  
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")  
[private]
```

Definiert in Zeile 43 der Datei QueryEditorBean.java.

Wird benutzt von QueryEditorBean.executeQuery(), QueryEditorBean.getQueryString() und QueryEditorBean.saveQuery().

6.1.4 QueryResultBean Klassenreferenz

6.1.4.1 Ausführliche Beschreibung

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei QueryResultBean.java.

Öffentliche Methoden

- DataObjectList getDataObjectList ()
- String getFetchTime ()
- void setDataObjectList (DataObjectList dataObjectList)
- void setFetchTime (String fetchTime)

Private Attribute

- DataObjectList dataObjectList
- String fetchTime

6.1.4.2 Dokumentation der Elementfunktionen

DataObjectList getDataObjectList ()

Definiert in Zeile 17 der Datei QueryResultBean.java.

Benutzt QueryResultBean.dataObjectList.

String getFetchTime ()

Definiert in Zeile 25 der Datei QueryResultBean.java.

Benutzt QueryResultBean.fetchTime.

void setDataObjectList (DataObjectList dataObjectList)

Definiert in Zeile 21 der Datei QueryResultBean.java.

void setFetchTime (String fetchTime)

Definiert in Zeile 29 der Datei QueryResultBean.java.

6.1.4.3 Dokumentation der Datenelemente**DataObjectList dataObjectList [private]**

Definiert in Zeile 14 der Datei QueryResultBean.java.

Wird benutzt von QueryResultBean.getDataObjectList().

String fetchTime [private]

Definiert in Zeile 15 der Datei QueryResultBean.java.

Wird benutzt von QueryResultBean.getFetchTime().

6.2 Tag-Library

6.2.1 DataObjectScrollerTag Klassenreferenz

6.2.1.1 Ausführliche Beschreibung

Das DataObjectScrollerTag stellt einen Scroller für dargestellte DataObjects durch das DataObjectTableTag dar.

Autor:

Roland Koppe

Definiert in Zeile 13 der Datei DataObjectScrollerTag.java.

Statische private Attribute

- static final long serialVersionUID = 1L

6.2.1.2 Dokumentation der Datenelemente

final long serialVersionUID = 1L [static, private]

Definiert in Zeile 15 der Datei DataObjectScrollerTag.java.

6.2.2 DataObjectTable Klassenreferenz

Abgeleitet von javax:servlet:jsp::tagext::TagSupport.

6.2.2.1 Ausführliche Beschreibung

Definiert in Zeile 20 der Datei DataObjectTable.java.

Öffentliche Methoden

- `int doAfterBody ()` throws `JspException`
- `int doEndTag ()` throws `JspException`
- `int doStartTag ()` throws `JspException`
- `String getTableId ()`
- `void setData (LinkedList data)`
- `void setPageContext (PageContext pageContext)`
- `void setTableId (String id)`

Private Attribute

- `LinkedList< DataObject > dataObjects = new LinkedList<DataObject>()`
- `String password =`
`(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")`
- `String tableId = ""`
- `String userId =`
`(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")`

Statische private Attribute

- `static final String linkEditInstanceObject =`
`"../extendDataInput/show_ObjectInstance.jsf?"`
- `static final String linkEditInstanceRelation =`
`"../extendDataInput/show_RelationInstance.jsf?"`
- `static final String linkMetaObjectId = "&metaobjectId="`
- `static final String linkMetaRelationId = "&metarelationId="`
- `static final String linkObjectId = "&instanceOfMetaobject="`
- `static final String linkRelationId = "&instanceOfMetaobject="`
- `static final long serialVersionUID = 1L`

6.2.2.2 Dokumentation der Elementfunktionen

`int doAfterBody ()` throws `JspException`

Definiert in Zeile 66 der Datei DataObjectTable.java.

int doEndTag () throws JspException

Definiert in Zeile 73 der Datei DataObjectTable.java.

int doStartTag () throws JspException

Gibt eine Tabelle mit Instanzen aus.

Autor:

Roland Koppe

Definiert in Zeile 85 der Datei DataObjectTable.java.

Benutzt DataObjectTable.dataObjects, DataObjectTable.linkEditInstanceObject, DataObjectTable.linkEditInstanceRelation, DataObjectTable.linkMetaObjectId, DataObjectTable.linkMetaRelationId, DataObjectTable.linkObjectId, DataObjectTable.linkRelationId, DataObjectTable.password, DataObjectTable.tableId und DataObjectTable.userId.

String getTableId ()

Definiert in Zeile 44 der Datei DataObjectTable.java.

Benutzt DataObjectTable.tableId.

void setData (LinkedList data)

Definiert in Zeile 54 der Datei DataObjectTable.java.

Benutzt DataObjectTable.dataObjects.

void setPageContext (PageContext pageContext)

Definiert in Zeile 235 der Datei DataObjectTable.java.

void setTableId (String id)

Definiert in Zeile 47 der Datei DataObjectTable.java.

Benutzt DataObjectTable.tableId.

6.2.2.3 Dokumentation der Datenelemente

```
LinkedList<DataObject> dataObjects = new LinkedList<DataObject>() [private]
```

Definiert in Zeile 38 der Datei DataObjectTable.java.

Wird benutzt von DataObjectTable.doStartTag() und DataObjectTable.setData().

```
final String linkEditInstanceObject =  
"..../extendDataInput/show_ObjectInstance.jsf?" [static, private]
```

Definiert in Zeile 29 der Datei DataObjectTable.java.

Wird benutzt von DataObjectTable.doStartTag().

```
final String linkEditInstanceRelation =  
"..../extendDataInput/show_RelationInstance.jsf?" [static, private]
```

Definiert in Zeile 33 der Datei DataObjectTable.java.

Wird benutzt von DataObjectTable.doStartTag().

```
final String linkMetaObjectId = "&metaobjectId=" [static, private]
```

Definiert in Zeile 27 der Datei DataObjectTable.java.

Wird benutzt von DataObjectTable.doStartTag().

```
final String linkMetaRelationId = "&metarelationId=" [static, private]
```

Definiert in Zeile 31 der Datei DataObjectTable.java.

Wird benutzt von DataObjectTable.doStartTag().

```
final String linkObjectId = "&instanceOfMetaobject=" [static, private]
```

Definiert in Zeile 28 der Datei DataObjectTable.java.

Wird benutzt von DataObjectTable.doStartTag().

```
final String linkRelationId = "&instanceOfMetaobject=" [static, private]
```

Definiert in Zeile 32 der Datei DataObjectTable.java.

Wird benutzt von DataObjectTable.doStartTag().

```
String password =  
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")  
[private]
```

Definiert in Zeile 25 der Datei DataObjectTable.java.

Wird benutzt von DataObjectTable.doStartTag().

```
final long serialVersionUID = 1L [static, private]
```

Definiert in Zeile 22 der Datei DataObjectTable.java.

```
String tableId = "" [private]
```

Definiert in Zeile 40 der Datei DataObjectTable.java.

Wird benutzt von DataObjectTable.doStartTag(), DataObjectTable.getTableId() und DataObjectTable.setTableId().

```
String userId =  
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")  
[private]
```

Definiert in Zeile 24 der Datei DataObjectTable.java.

Wird benutzt von DataObjectTable.doStartTag().

6.2.3 DataObjectTableTag Klassenreferenz

6.2.3.1 Ausführliche Beschreibung

Dieses Tag zeigt eine DataObjectList in einer Tabelle an.

Autor:

Roland Koppe

Definiert in Zeile 27 der Datei DataObjectTableTag.java.

Öffentliche Methoden

- int doStartTag () throws JspException

Private Attribute

- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Statische private Attribute

- static final String linkEditInstanceObject =
"/extendDataInput/show_ObjectInstance.jsf?"
- static final String linkEditInstanceRelation =
"/extendDataInput/show_RelationInstance.jsf?"
- static final String linkMetaObjectId = "&metaobjectId="
- static final String linkMetaRelationId = "&metarelationId="

- static final String linkObjectId = "&instanceOfMetaobject="
- static final String linkRelationId = "&instanceOfMetaobject="
- static final long serialVersionUID = 1L

6.2.3.2 Dokumentation der Elementfunktionen

int doStartTag () throws JspException

Definiert in Zeile 45 der Datei DataObjectTableTag.java.

Benutzt DataObjectTableTag.linkEditInstanceObject, DataObjectTableTag.linkEditInstanceRelation, DataObjectTableTag.linkMetaObjectId, DataObjectTableTag.linkMetaRelationId, DataObjectTableTag.linkObjectId, DataObjectTableTag.linkRelationId, DataObjectTableTag.password und DataObjectTableTag.userId.

6.2.3.3 Dokumentation der Datenelemente

final String linkEditInstanceObject =
"../extendDataInput/show_ObjectInstance.jsf?" [static, private]

Definiert in Zeile 36 der Datei DataObjectTableTag.java.

Wird benutzt von DataObjectTableTag.doStartTag().

final String linkEditInstanceRelation =
"../extendDataInput/show_RelationInstance.jsf?" [static, private]

Definiert in Zeile 40 der Datei DataObjectTableTag.java.

Wird benutzt von DataObjectTableTag.doStartTag().

final String linkMetaObjectId = "&metaobjectId=" [static, private]

Definiert in Zeile 34 der Datei DataObjectTableTag.java.

Wird benutzt von DataObjectTableTag.doStartTag().

final String linkMetaRelationId = "&metarelationId=" [static, private]

Definiert in Zeile 38 der Datei DataObjectTableTag.java.

Wird benutzt von DataObjectTableTag.doStartTag().

final String linkObjectId = "&instanceOfMetaobject=" [static, private]

Definiert in Zeile 35 der Datei DataObjectTableTag.java.

Wird benutzt von DataObjectTableTag.doStartTag().

final String linkRelationId = "&instanceOfMetaobject=" [static, private]

Definiert in Zeile 39 der Datei DataObjectTableTag.java.

Wird benutzt von DataObjectTableTag.doStartTag().

String password =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
[private]

Definiert in Zeile 32 der Datei DataObjectTableTag.java.

Wird benutzt von DataObjectTableTag.doStartTag().

final long serialVersionUID = 1L [static, private]

Definiert in Zeile 29 der Datei DataObjectTableTag.java.

String userId =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")
[private]

Definiert in Zeile 31 der Datei DataObjectTableTag.java.

Wird benutzt von DataObjectTableTag.doStartTag().

7 de.offis.pg.eam.mod_exp_jasper

7.1 BundleConfiguration

7.1.1 BundleViews Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::module::api::IModuleView`.

7.1.1.1 Ausführliche Beschreibung

Definiert in Zeile 21 der Datei `BundleViews.java`.

Öffentliche Methoden

- `String getBundleVersion ()`
- `String getDescription ()`
- `long getEquinoxBundleId ()`
- `String getName ()`
- `List< View > getViews ()`

Statische private Attribute

- `static final String VIEWS = "bundle_views.xml"`

7.1.1.2 Dokumentation der Elementfunktionen

String getBundleVersion ()

Definiert in Zeile 83 der Datei `BundleViews.java`.

String getDescription ()

Definiert in Zeile 28 der Datei `BundleViews.java`.

long getEquinoxBundleId ()

Definiert in Zeile 34 der Datei `BundleViews.java`.

String getName ()

Definiert in Zeile 40 der Datei BundleViews.java.

List<View> getViews ()

Definiert in Zeile 46 der Datei BundleViews.java.

Benutzt BundleViews.VIEWS.

7.1.1.3 Dokumentation der Datenelemente

final String VIEWS = "bundle_views.xml" [static, private]

Definiert in Zeile 24 der Datei BundleViews.java.

Wird benutzt von BundleViews.getViews().

7.1.2 Configuration Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::configuration::api::IBundleConfiguration`.

7.1.2.1 Ausführliche Beschreibung

Definiert in Zeile 10 der Datei Configuration.java.

Öffentliche Methoden

- `boolean bundleHasObjects ()`
- `boolean bundleHasRelations ()`
- `boolean bundleHasViews ()`
- `Document getObjectsJdomDocument ()` throws `JDOMException`, `IOException`

7.1.2.2 Dokumentation der Elementfunktionen

boolean bundleHasObjects ()

Definiert in Zeile 13 der Datei Configuration.java.

boolean bundleHasRelations ()

Definiert in Zeile 28 der Datei Configuration.java.

boolean bundleHasViews ()

Definiert in Zeile 18 der Datei Configuration.java.

Document `getObjectsJdomDocument ()` throws **JDOMEException**, **IOException**

Definiert in Zeile 23 der Datei `Configuration.java`.

7.1.3 BundleProperties Klassenreferenz

7.1.3.1 Ausführliche Beschreibung

Definiert in Zeile 11 der Datei `BundleProperties.java`.

Öffentliche, statische Methoden

- `static void load ()`

Private Typen

- `enum Property {`
 `DB_DRIVER_CLASS, DB_SCHEMA, DB_SERVER, DB_USER,`
 `DB_PASSWORD, DB_DATABASE }`

Private, statische Methoden

- `static void initStandards ()`

Statische private Attribute

- `static Properties properties = new Properties()`
- `static String propertyFile = System.getProperty("user.dir") + "/eamconfig" +`
 `"/mod_export.properties"`
- `static Properties standards = new Properties()`

7.1.3.2 Dokumentation der Aufzählungstypen

`enum Property [private]`

Aufzählungswerte:

DB_DRIVER_CLASS
DB_SCHEMA
DB_SERVER
DB_USER
DB_PASSWORD

DB_DATABASE

Definiert in Zeile 13 der Datei BundleProperties.java.

7.1.3.3 Dokumentation der Elementfunktionen

static void initStandards () [static, private]

Definiert in Zeile 21 der Datei BundleProperties.java.

Benutzt BundleProperties.standards.

Wird benutzt von BundleProperties.load().

static void load () [static]

Definiert in Zeile 32 der Datei BundleProperties.java.

Benutzt BundleProperties.initStandards(), BundleProperties.properties, BundleProperties.propertyFile und BundleProperties.standards.

7.1.3.4 Dokumentation der Datenelemente

Properties properties = new Properties() [static, private]

Definiert in Zeile 17 der Datei BundleProperties.java.

Wird benutzt von BundleProperties.load().

**String propertyFile = System.getProperty("user.dir") + "/eamconfig" +
"/mod_export.properties"** [static, private]

Definiert in Zeile 15 der Datei BundleProperties.java.

Wird benutzt von BundleProperties.load().

Properties standards = new Properties() [static, private]

Definiert in Zeile 18 der Datei BundleProperties.java.

Wird benutzt von BundleProperties.initStandards() und BundleProperties.load().

7.1.4 Activator Klassenreferenz

Abgeleitet von org::osgi::framework::BundleActivator.

7.1.4.1 Ausführliche Beschreibung

Definiert in Zeile 57 der Datei Activator.java.

Öffentliche Methoden

- void start (BundleContext context) throws Exception
- void stop (BundleContext context) throws Exception

Statische öffentliche Attribute

- static String bundleSymbolicName
- static String bundleVersion
- static long equinoxBundleId
- static final String PATH = "/mod_export"

Private Attribute

- ServiceTracker httpServiceTracker

Statische private Attribute

- static final String WEB_ROOT = "/web"

Klassen

- class HttpServiceTracker
- class ServletContextListenerServletAdaptor

7.1.4.2 Dokumentation der Elementfunktionen

void start (BundleContext context) throws Exception

Definiert in Zeile 71 der Datei Activator.java.

Benutzt Activator.bundleSymbolicName, Activator.bundleVersion, Activator.equinoxBundleId und Activator.httpServiceTracker.

void stop (BundleContext context) throws Exception

Definiert in Zeile 91 der Datei Activator.java.

Benutzt Activator.httpServiceTracker.

7.1.4.3 Dokumentation der Datenelemente

String bundleSymbolicName [static]

Definiert in Zeile 67 der Datei Activator.java.

Wird benutzt von Activator.start().

String bundleVersion [static]

Definiert in Zeile 68 der Datei Activator.java.

Wird benutzt von Activator.start().

long equinoxBundleId [static]

Definiert in Zeile 69 der Datei Activator.java.

Wird benutzt von Activator.start().

ServiceTracker httpServiceTracker [private]

Definiert in Zeile 65 der Datei Activator.java.

Wird benutzt von Activator.start() und Activator.stop().

final String PATH = "/mod_export" [static]

Definiert in Zeile 60 der Datei Activator.java.

Wird benutzt von Activator.HttpServiceTracker.addingService() und Activator.HttpServiceTracker.removedService().

final String WEB_ROOT = "/web" [static, private]

Definiert in Zeile 62 der Datei Activator.java.

Wird benutzt von Activator.HttpServiceTracker.addingService().

7.1.5 Activator.HttpServiceTracker Klassenreferenz

Abgeleitet von org::osgi::util::tracker::ServiceTracker.

7.1.5.1 Ausführliche Beschreibung

Definiert in Zeile 99 der Datei Activator.java.

Öffentliche Methoden

- Object addingService (ServiceReference reference)
- HttpServiceTracker (BundleContext context)
- void removedService (ServiceReference reference, Object service)

7.1.5.2 Beschreibung der Konstruktoren und Destruktoren

HttpServiceTracker (BundleContext context)

Definiert in Zeile 107 der Datei Activator.java.

7.1.5.3 Dokumentation der Elementfunktionen

Object addingService (ServiceReference reference)

Definiert in Zeile 115 der Datei Activator.java.

Benutzt Activator.PATH und Activator.WEB_ROOT.

void removedService (ServiceReference reference, Object service)

Definiert in Zeile 180 der Datei Activator.java.

Benutzt Activator.PATH.

7.1.6 Activator.ServletContextListenerServletAdaptor Klassenreferenz

Abgeleitet von javax::servlet::Servlet.

7.1.6.1 Ausführliche Beschreibung

Definiert in Zeile 198 der Datei Activator.java.

Öffentliche Methoden

- void destroy ()
- ServletConfig getServletConfig ()
- String getServletInfo ()
- void init (ServletConfig config) throws ServletException
- void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException
- ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Private Attribute

- ServletConfig config
- Servlet delegate
- ClassLoader jspLoader

- ServletContextListener listener

7.1.6.2 Beschreibung der Konstruktoren und Destruktoren

ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Definiert in Zeile 210 der Datei Activator.java.

7.1.6.3 Dokumentation der Elementfunktionen

void destroy ()

Definiert in Zeile 278 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config, Activator.ServletContextListenerServletAdaptor.delegate, Activator.ServletContextListenerServletAdaptor.jspLoader und Activator.ServletContextListenerServletAdaptor.listener.

ServletConfig getServletConfig ()

Definiert in Zeile 306 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config.

String getServletInfo ()

Definiert in Zeile 314 der Datei Activator.java.

void init (ServletConfig config) throws ServletException

Definiert in Zeile 226 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config, Activator.ServletContextListenerServletAdaptor.delegate, Activator.ServletContextListenerServletAdaptor.jspLoader und Activator.ServletContextListenerServletAdaptor.listener.

void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException

Definiert in Zeile 254 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.delegate und Activator.ServletContextListenerServletAdaptor.jspLoader.

7.1.6.4 Dokumentation der Datenelemente

ServletConfig config [private]

Definiert in Zeile 200 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(), Activator.ServletContextListenerServletAdaptor.getServletConfig() und Activator.ServletContextListenerServletAdaptor.init().

Servlet delegate [private]

Definiert in Zeile 204 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(), Activator.ServletContextListenerServletAdaptor.init() und Activator.ServletContextListenerServletAdaptor.service().

ClassLoader jspLoader [private]

Definiert in Zeile 206 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(), Activator.ServletContextListenerServletAdaptor.init() und Activator.ServletContextListenerServletAdaptor.service().

ServletContextListener listener [private]

Definiert in Zeile 202 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy() und Activator.ServletContextListenerServletAdaptor.init().

7.1.7 TopMenuBar Klassenreferenz

7.1.7.1 Ausführliche Beschreibung

Definiert in Zeile 8 der Datei TopMenuBar.java.

Öffentliche, statische Methoden

- static TopMenuBar getInstance ()

Geschützte Methoden

- void activate (ComponentContext context)
- void deactivate (ComponentContext context) throws Exception

Statische Paketattribute

- static TopMenuBar instance

7.1.7.2 Dokumentation der Elementfunktionen

void activate (ComponentContext context) [protected]

Definiert in Zeile 13 der Datei TopMenuBar.java.

Benutzt TopMenuBar.instance.

void deactivate (ComponentContext context) throws Exception [protected]

Definiert in Zeile 18 der Datei TopMenuBar.java.

static TopMenuBar getInstance () [static]

Definiert in Zeile 22 der Datei TopMenuBar.java.

Benutzt TopMenuBar.instance.

7.1.7.3 Dokumentation der Datenelemente

TopMenuBar instance [static, package]

Definiert in Zeile 10 der Datei TopMenuBar.java.

Wird benutzt von TopMenuBar.activate() und TopMenuBar.getInstance().

7.1.8 Menu Klassenreferenz

Abgeleitet von *de::offis::pg::eam::core::menu::api::AbstractAdoptableMenuContributor*.

7.1.8.1 Ausführliche Beschreibung

Definiert in Zeile 6 der Datei Menu.java.

Öffentliche Methoden

- String getAdopterId ()
- String getContributionTarget ()
- String getContributorId ()
- String getTargetSubmenu ()
- Menu ()

7.1.8.2 Beschreibung der Konstruktoren und Destruktoren

Menu ()

Definiert in Zeile 8 der Datei Menu.java.

7.1.8.3 Dokumentation der Elementfunktionen

String getAdopterId ()

Definiert in Zeile 12 der Datei Menu.java.

String getContributionTarget ()

Definiert in Zeile 16 der Datei Menu.java.

String getContributorId ()

Definiert in Zeile 20 der Datei Menu.java.

String getTargetSubmenu ()

Definiert in Zeile 24 der Datei Menu.java.

7.2 Datenbank

7.2.1 Connection Klassenreferenz

7.2.1.1 Ausführliche Beschreibung

Stellt die Verbindung zur Datenbank bereit. Die Standard-Einstellungen für die Verbindung sollen mit `Connection.setup()` gesetzt werden. Nach der erfolgreichen Verbindung kann mit `Connection.getConnection()` die Verbindung verwendet werden.

Autor:

Roland Koppe

Definiert in Zeile 26 der Datei Connection.java.

Öffentliche, statische Methoden

- static synchronized void close ()
- static synchronized boolean connect () throws SQLException, DatabaseConnectionException
- static java.sql.Connection getConnection () throws DatabaseException

- static String getDatabase ()
- static Driver getDbDriver ()
- static boolean isClosed ()
- static synchronized void setDatabase (String database)
- static void setDbDriverClass (String driver)
- static void setup (String driver, String dbSchema, String dbServer, String dbUser, String dbPassword)
- static void setup (String dbSchema, String dbServer, String dbUser, String dbPassword)

Private, statische Methoden

- static synchronized boolean loadDbDriver () throws DatabaseConnectionException

Statische private Attribute

- static java.sql.Connection connection = null
- static ConnectionHolder connectionHolder = null
- static String database = ""
- static com.mysql.jdbc.Driver dbDriver = null
- static String dbDriverClass = ""
- static String dbPassword = ""
- static int dbRetryConnectionCount = 3
- static String dbSchema = ""
- static String dbServer = ""
- static String dbUser = ""

7.2.1.2 Dokumentation der Elementfunktionen

static synchronized void close () [static]

Schließt die Datenbankverbindung.

Definiert in Zeile 198 der Datei Connection.java.

Benutzt Connection.close() und Connection.connection.

Wird benutzt von Connection.close().

static synchronized boolean connect () throws SQLException, DatabaseConnectionException [static]

Stellt die Verbindung zur Datenbank her.

Ausnahmebehandlung:

SQLException, wenn SQL Fehler

Rückgabe:

true, wenn erfolgreich, sonst false

Ausnahmebehandlung:

DatabaseConnectionException

Definiert in Zeile 125 der Datei Connection.java.

Benutzt Connection.connection, Connection.connectionHolder, Connection.database, Connection.dbPassword, Connection.dbSchema, Connection.dbServer, Connection.dbUser, Connection.getConnection(), Connection.loadDbDriver() und Connection.setDatabase().

Wird benutzt von Connection.getConnection().

static java.sql.Connection getConnection () throws DatabaseException [static]

Liefert die Datenbankverbindung.

Ausnahmebehandlung:

DatabaseException

Rückgabe:

java.sql.Connection, sonst null

Definiert in Zeile 222 der Datei Connection.java.

Benutzt Connection.connect(), Connection.connection und Connection.dbRetryConnectionCount.

Wird benutzt von Connection.connect().

static String getDatabase () [static]

Liefert den Namen des Katalogs der Verbindung, wenn dieser zuvor gesetzt worden ist.

Rückgabe:

String

Definiert in Zeile 296 der Datei Connection.java.

Benutzt Connection.database.

static Driver getDbDriver () [static]

Liefert den Treiber für die Datenbankverbindung.

Rückgabe:

Driver

Definiert in Zeile 262 der Datei Connection.java.

Benutzt Connection.dbDriver.

static boolean isClosed () [static]

Gibt zurück, ob die Datenbankverbindung geschlossen ist.

Rückgabe:

true, Verbindung geschlossen

Definiert in Zeile 178 der Datei Connection.java.

Benutzt Connection.connection und Connection.isClosed().

Wird benutzt von Connection.isClosed().

**static synchronized boolean loadDbDriver () throws DatabaseConnectionException
[static, private]**

Lädt den gesetzten Datenbanktreiber.

Rückgabe:

true, wenn Treiber geladen, sonst false

Ausnahmebehandlung:

DatabaseConnectionException

Definiert in Zeile 104 der Datei Connection.java.

Benutzt Connection.dbDriver und Connection.dbDriverClass.

Wird benutzt von Connection.connect().

static synchronized void setDatabase (String database) [static]

Setzt den Katalog für die Datenbankverbindung.

Parameter:

database String

Definiert in Zeile 273 der Datei Connection.java.

Benutzt Connection.connection.

Wird benutzt von Connection.connect().

static void setDbDriverClass (String driver) [static]

Setzt den Treiber für die Datenbankverbindung als qualifizierten Namen der Klasse. Beispielsweise kann die Klasse com.mysql.jdbc.Driver benutzt werden.

Parameter:

driver String com.mysql.jdbc.Driver

Definiert in Zeile 59 der Datei Connection.java.

Wird benutzt von Connection.setup().

static void setup (String driver, String dbSchema, String dbServer, String dbUser, String dbPassword) [static]

Setzt die Standardeinstellungen für die Verbindung.

Parameter:

driver String com.mysql.jdbc.Driver

dbSchema String Schema jdbc:mysql:

dbServer String Server 134.106.56.251:3307

dbUser String Benutzername

dbPassword String Passwort

Definiert in Zeile 91 der Datei Connection.java.

Benutzt Connection.setDbDriverClass() und Connection.setup().

static void setup (String dbSchema, String dbServer, String dbUser, String dbPassword) [static]

Setzt die Standardeinstellungen für die Verbindung.

Parameter:

dbSchema String Schema jdbc:mysql:

dbServer String Server 134.106.56.251:3307

dbUser String Benutzername

dbPassword String Passwort

Definiert in Zeile 73 der Datei Connection.java.

Wird benutzt von Connection.setup().

7.2.1.3 Dokumentation der Datenelemente

java.sql.Connection connection = null [static, private]

Definiert in Zeile 46 der Datei Connection.java.

Wird benutzt von `Connection.close()`, `Connection.connect()`, `Connection.getConnection()`, `Connection.isClosed()` und `Connection.setDatabase()`.

ConnectionHolder connectionHolder = null [static, private]

Definiert in Zeile 48 der Datei `Connection.java`.

Wird benutzt von `Connection.connect()`.

String database = "" [static, private]

Definiert in Zeile 43 der Datei `Connection.java`.

Wird benutzt von `Connection.connect()` und `Connection.getDatabase()`.

com.mysql.jdbc.Driver dbDriver = null [static, private]

Definiert in Zeile 45 der Datei `Connection.java`.

Wird benutzt von `Connection.getDbDriver()` und `Connection.loadDbDriver()`.

String dbDriverClass = "" [static, private]

Definiert in Zeile 38 der Datei `Connection.java`.

Wird benutzt von `Connection.loadDbDriver()`.

String dbPassword = "" [static, private]

Definiert in Zeile 42 der Datei `Connection.java`.

Wird benutzt von `Connection.connect()`.

int dbRetryConnectionCount = 3 [static, private]

Definiert in Zeile 29 der Datei `Connection.java`.

Wird benutzt von `Connection.getConnection()`.

String dbSchema = "" [static, private]

Definiert in Zeile 39 der Datei `Connection.java`.

Wird benutzt von `Connection.connect()`.

String dbServer = "" [static, private]

Definiert in Zeile 40 der Datei `Connection.java`.

Wird benutzt von `Connection.connect()`.

String dbUser = "" [static, private]

Definiert in Zeile 41 der Datei Connection.java.

Wird benutzt von Connection.connect().

7.2.2 ConnectionHolder Klassenreferenz

7.2.2.1 Ausführliche Beschreibung

Der ConnectionHolder sendet in angegebenen Abständen eine Anfrage an die Datenbank, um die Verbindung aufrecht zu erhalten.

Autor:

Roland Koppe

Definiert in Zeile 16 der Datei ConnectionHolder.java.

Öffentliche Methoden

- void run ()

Öffentliche, statische Methoden

- static long getWaittime ()
- static void setWaittime (String waittime)

Statische private Attribute

- static final long STANDARD_WAITTIME = 5 * 60
- static long waittime = STANDARD_WAITTIME * 1000

7.2.2.2 Dokumentation der Elementfunktionen

static long getWaittime () [static]

Liefert die Wartezeit bis zum Senden der nächsten Anfrage in Sekunden.

Rückgabe:

long Wartezeit in Sekunden

Definiert in Zeile 51 der Datei ConnectionHolder.java.

Benutzt ConnectionHolder.waittime.

void run ()

Startet den ConnectionHolder.

Definiert in Zeile 26 der Datei ConnectionHolder.java.

Benutzt ConnectionHolder.waittime.

static void setWaittime (String waittime) [static]

Setzt die Wartezeit in Sekunden bis zur nächsten Anfrage.

Parameter:

waittime String Wartezeit in Sekunden

Definiert in Zeile 63 der Datei ConnectionHolder.java.

Benutzt ConnectionHolder.STANDARD_WAITTIME.

7.2.2.3 Dokumentation der Datenelemente

final long STANDARD_WAITTIME = 5 * 60 [static, private]

Definiert in Zeile 18 der Datei ConnectionHolder.java.

Wird benutzt von ConnectionHolder.setWaittime().

long waittime = STANDARD_WAITTIME * 1000 [static, private]

Definiert in Zeile 19 der Datei ConnectionHolder.java.

Wird benutzt von ConnectionHolder.getWaittime() und ConnectionHolder.run().

7.2.3 Database Klassenreferenz

7.2.3.1 Ausführliche Beschreibung

Diese Klasse dient dem Ausführen von Anfragen an die Datenbank.

<http://dev.mysql.com/doc/refman/5.0/en/connector-j-usagenotes-troubleshooting.html#qandaitem-26-4-5-3-4>

<http://dev.mysql.com/doc/refman/5.1/en/error-messages-server.html>

Autor:

Roland Koppe

Definiert in Zeile 29 der Datei Database.java.

Öffentliche, statische Methoden

- static boolean createDatabase (String database) throws DatabaseException
- static boolean deleteDatabase (String database) throws DatabaseException
- static synchronized ArrayList< Integer > execute (String sql) throws DatabaseException
- static synchronized ResultSet executeQuery (String sql) throws DatabaseException
- static boolean existsTable (String table) throws DatabaseException
- static List< String > getDatabaseTables (String database) throws DatabaseException
- static void printResultSet (ResultSet rs)
- static void releaseQuery (ResultSet rs)
- static ArrayList< String > splitSQL (String sql)
- static String uncommentSQL (String sql)
- static boolean useDatabase (String database) throws DatabaseException

Statische Paketattribute

- static final int RETRY_CONNECT_COUNT = 3

7.2.3.2 Dokumentation der Elementfunktionen

static boolean createDatabase (String database) throws DatabaseException

[static]

Erstellt eine Datenbankschema.

Parameter:

database String Namen des Schema

Rückgabe:

true, wenn erfolgreich

Ausnahmebehandlung:

DatabaseException

Definiert in Zeile 348 der Datei Database.java.

Benutzt Database.execute().

static boolean deleteDatabase (String database) throws DatabaseException

[static]

Löscht ein Datenbankschema.

Parameter:

database String Name des Schema , wenn erfolgreich

Ausnahmebehandlung:

DatabaseException

Definiert in Zeile 361 der Datei Database.java.

Benutzt Database.execute().

static synchronized ArrayList<Integer> execute (String sql) throws DatabaseException [static]

Führt eine oder mehrere durch Semikolon getrennte SQL Anfragen aus und liefert ein Array mit Rückgaben für jede Anfrage.

Folgende Werte kann das Array enthalten: -1 die Anfrage war nicht erfolgreich (gibt es nicht wirklich. Es werden vorher Exceptions geworfen) 0 die Anfrage war erfolgreich, erzeugte aber kein auto_inc Wert >0 die Anfrage erzeugte einen neuen Datensatz mit der ID als Rückgabewert.

Nutzung für DDL, INSERT, UPDATE und DELETE.

Parameter:

sql SQL Anfrage(n) getrennt durch Semikolon.

Ausnahmebehandlung:

DatabaseException

Rückgabe:

-1 Anfrage nicht erfolgreich, 0 Anfrage erfolgreich, > 0 ID eines neuen Datensatzs

Definiert in Zeile 131 der Datei Database.java.

Benutzt Database.splitSQL() und Database.uncommentSQL().

Wird benutzt von Database.createDatabase(), Database.deleteDatabase() und Database.useDatabase().

static synchronized ResultSet executeQuery (String sql) throws DatabaseException [static]

Definiert in Zeile 262 der Datei Database.java.

Wird benutzt von Database.existsTable() und Database.getDatabaseTables().

static boolean existsTable (String table) throws DatabaseException [static]

Gibt zurück, ob eine Tabelle existiert oder nicht.

Parameter:

table String Name der Tabelle

Rückgabe:

true, wenn die Tabelle existiert

Ausnahmebehandlung:**DatabaseException**

Definiert in Zeile 417 der Datei Database.java.

Benutzt Database.executeQuery().

static List<String> getDatabaseTables (String database) throws DatabaseException
[static]

Liefert ein ResultSet zurück, welches alle Tabellen eines Schema enthält.

Parameter:

database String Name des Schema

Rückgabe:

ResultSet Liste von Tabellen

Ausnahmebehandlung:**DatabaseException**

Definiert in Zeile 388 der Datei Database.java.

Benutzt Database.executeQuery().

static void printResultSet (ResultSet rs) [static]

Definiert in Zeile 437 der Datei Database.java.

static void releaseQuery (ResultSet rs) [static]

Definiert in Zeile 319 der Datei Database.java.

static ArrayList<String> splitSQL (String sql) [static]

Trennt eine gegebene SQL Zeichenfolge an den Semikolon (;), solange nicht das Kommando DELIMITER gesetzt wird. Ist dieses gesetzt wird die Zeichenfolge an diesem DELIMITER getrennt.

Parameter:

sql String

Rückgabe:

List<String>

Definiert in Zeile 43 der Datei Database.java.

Wird benutzt von Database.execute().

static String uncommentSQL (String sql) [static]

Entfernt SQL Kommentare beginnend mit "--" aus dem angegebenen SQL Skript.

Parameter:

sql String SQL Skript

Rückgabe:

String SQL Skript ohne Kommentare (–)

Definiert in Zeile 107 der Datei Database.java.

Wird benutzt von Database.execute().

static boolean useDatabase (String database) throws DatabaseException [static]

Wählt das Standard-Schema der Datenbank aus.

Parameter:

database String Name des Schema

Rückgabe:

true, wenn das Schema existiert

Ausnahmebehandlung:

DatabaseException

Definiert in Zeile 374 der Datei Database.java.

Benutzt Database.execute().

7.2.3.3 Dokumentation der Datenelemente

final int RETRY_CONNECT_COUNT = 3 [static, package]

Definiert in Zeile 31 der Datei Database.java.

7.2.4 Design Klassenreferenz

7.2.4.1 Ausführliche Beschreibung

Definiert in Zeile 10 der Datei Design.java.

Öffentliche Methoden

- String getDescription ()
- JRField[] getFields ()
- String getFilename ()
- JRParameter[] getParameters ()
- JRVariable[] getVars ()
- void setDescription (String description)
- void setFields (JRField[] fields)
- void setFilename (String filename)
- void setParameters (JRParameter[] parameters)
- void setVars (JRVariable[] vars)

Private Attribute

- String description
- JRField[] fields
- String filename
- JRParameter[] parameters
- JRVariable[] vars

7.2.4.2 Dokumentation der Elementfunktionen

String getDescription ()

Definiert in Zeile 16 der Datei Design.java.

Benutzt Design.description.

JRField [] getFields ()

Definiert in Zeile 28 der Datei Design.java.

Benutzt Design.fields.

Wird benutzt von Report.createData() und ReportFrontend.getFieldItems().

String getFilename ()

Definiert in Zeile 22 der Datei Design.java.

Benutzt Design.filename.

JRParameter [] getParameters ()

Definiert in Zeile 34 der Datei Design.java.

Benutzt Design.parameters.

JRVariable [] getVars ()

Definiert in Zeile 40 der Datei Design.java.

Benutzt Design.vars.

void setDescription (String description)

Definiert in Zeile 19 der Datei Design.java.

void setFields (JRField[] fields)

Definiert in Zeile 31 der Datei Design.java.

Wird benutzt von Report.createDesignObject().

void setFilename (String filename)

Definiert in Zeile 25 der Datei Design.java.

Wird benutzt von Report.createDesignObject().

void setParameters (JRParameter[] parameters)

Definiert in Zeile 37 der Datei Design.java.

Wird benutzt von Report.createDesignObject().

void setVars (JRVariable[] vars)

Definiert in Zeile 43 der Datei Design.java.

Wird benutzt von Report.createDesignObject().

7.2.4.3 Dokumentation der Datenelemente

String description [private]

Definiert in Zeile 11 der Datei Design.java.

Wird benutzt von Design.getDescription().

JRField [] fields [private]

Definiert in Zeile 13 der Datei Design.java.

Wird benutzt von Design.getFields().

String filename [private]

Definiert in Zeile 12 der Datei Design.java.

Wird benutzt von Design.getFilename().

JRParameter [] parameters [private]

Definiert in Zeile 14 der Datei Design.java.

Wird benutzt von Design.getParameters().

JRVariable [] vars [private]

Definiert in Zeile 15 der Datei Design.java.

Wird benutzt von Design.getVars().

7.3 Texteditor

7.3.1 TextEditor Klassenreferenz

7.3.1.1 Ausführliche Beschreibung

Text Generator Klasse für das dynmische Erzeugen von Berichten

Autor:

Mart Koehler

Definiert in Zeile 30 der Datei TextEditor.java.

Öffentliche Methoden

- void addQueries (Long id, String name)
- void addText (String optionName, String text)
- ArrayList< HashMap > buildData (ArrayList< String > options)
- ArrayList< String > getAllNames ()
- ArrayList< String > getAllText ()

- `HashMap< Long, String > getQueries ()`
- `ArrayList< Long > getQueryIds ()`
- `ArrayList< String > getQueryNames ()`
- `LinkedList< Query > getQueryObjects ()`
- `String getSingleText (String optionName)`
- `boolean removeText (String option)`
- `void setQueries (HashMap< Long, String > queries)`
- `TextEditor ()`

Private Attribute

- `HashMap< Long, String > queries`

7.3.1.2 Beschreibung der Konstruktoren und Destruktoren

TextEditor ()

Definiert in Zeile 33 der Datei `TextEditor.java`.

Benutzt `TextEditor.getQueryObjects()`.

7.3.1.3 Dokumentation der Elementfunktionen

void addQueries (Long id, String name)

Definiert in Zeile 260 der Datei `TextEditor.java`.

Benutzt `TextEditor.queries`.

Wird benutzt von `TextEditor.getQueryObjects()`.

void addText (String optionName, String text)

Es wird ein Text zur Datenbank hinzugefügt, der in einem entsprechenden Berichtstemplate über option verwendet werden kann

Parameter:

option Gleichbedeutend mit einem Paramter Namen eines Templates

Text Der enthaltende Text mit Option als Platzhalter für customized Paramter

Definiert in Zeile 173 der Datei `TextEditor.java`.

ArrayList<HashMap> buildData (ArrayList< String > options)

Export gibt eine Liste von Optionen an, die er von einem Template hat und fordert die dahinterliegenden Daten an

Parameter:

options Liste von Optionen, die vom template kommt

Rückgabe:

Zusammengebaute HashMap passend für den Jasper Export

Definiert in Zeile 42 der Datei TextEditor.java.

Benutzt TextEditor.getQueryObjects().

Wird benutzt von Report.createData().

ArrayList<String> getAllNames ()

Definiert in Zeile 232 der Datei TextEditor.java.

ArrayList<String> getAllText ()

Alle Texte aus der Datenbank holen

Rückgabe:

ArrayList von Texten

Definiert in Zeile 212 der Datei TextEditor.java.

HashMap<Long, String> getQueries ()

Definiert in Zeile 252 der Datei TextEditor.java.

Benutzt TextEditor.queries.

Wird benutzt von TextEditor.getQueryIds() und TextEditor.getQueryNames().

ArrayList<Long> getQueryIds ()

Nur die IDs der Query Objekte holen

Rückgabe:

Liste von IDs

Definiert in Zeile 292 der Datei TextEditor.java.

Benutzt TextEditor.getQueries() und TextEditor.getQueryObjects().

ArrayList<String> getQueryNames ()

Nur Namen von Query Objekten holen

Rückgabe:

Liste von Namen

Definiert in Zeile 278 der Datei `TextEditor.java`.

Benutzt `TextEditor.getQueries()` und `TextEditor.getQueryObjects()`.

LinkedList<Query> getQueryObjects ()

Alle Query Objekte, die mit dem QueryEditor erstellt wurden holen

Rückgabe:

Liste mit Queries

Definiert in Zeile 307 der Datei `TextEditor.java`.

Benutzt `TextEditor.addQueries()` und `TextEditor.setQueries()`.

Wird benutzt von `TextEditor.buildData()`, `TextEditor.getQueryIds()`, `TextEditorBean.getQueryList()`, `TextEditor.getQueryNames()` und `TextEditor.TextEditor()`.

String getSingleText (String optionName)

Zu einem Optionsnamen den Text holen

Parameter:

optionName eindeutiger Optionsname

Rückgabe:

Text zu der Option

Definiert in Zeile 186 der Datei `TextEditor.java`.

boolean removeText (String option)

der Angegebene Text wird aus der Datenbank entfernt

Parameter:

option Die Option, über die der Text gefunden und entfernt werden kann

Rückgabe:

true: erfolgreich, false: fehlgeschlagen

Definiert in Zeile 269 der Datei `TextEditor.java`.

void setQueries (HashMap< Long, String > queries)

Definiert in Zeile 256 der Datei `TextEditor.java`.

Wird benutzt von `TextEditor.getQueryObjects()`.

7.3.1.4 Dokumentation der Datenelemente

HashMap<Long, String> queries [private]

Definiert in Zeile 31 der Datei TextEditor.java.

Wird benutzt von TextEditor.addQueries() und TextEditor.getQueries().

7.3.2 TextEditorBean Klassenreferenz

7.3.2.1 Ausführliche Beschreibung

Bean Klasse für den Text Editor Hier werden im Grunde nur Datenbankabfragen und manipulationen durchgeführt

Autor:

Mart Koehler

Definiert in Zeile 25 der Datei TextEditorBean.java.

Öffentliche Methoden

- void deleteText ()
- List getQueryList ()
- String getQuerySelect ()
- TextEditor getTextEditor ()
- List getTextItems ()
- String getTextNameSelect ()
- String getTextSelect ()
- void newText ()
- void saveText ()
- void setQuerySelect (String querySelect)
- void setTextNameSelect (String textNameSelect)
- void setTextSelect (String textSelect)
- void textareaChange (ValueChangeEvent valueChangeEvent)
- TextEditorBean ()

Private Attribute

- String querySelect
- TextEditor textEditor
- String textNameSelect
- String textSelect

7.3.2.2 Beschreibung der Konstruktoren und Destruktoren

TextEditorBean ()

Definiert in Zeile 55 der Datei TextEditorBean.java.

Benutzt TextEditorBean.textEditor.

7.3.2.3 Dokumentation der Elementfunktionen

void deleteText ()

Definiert in Zeile 144 der Datei TextEditorBean.java.

Benutzt TextEditorBean.getTextNameSelect(), TextEditorBean.setTextNameSelect() und TextEditorBean.setTextSelect().

List getQueryList ()

Definiert in Zeile 59 der Datei TextEditorBean.java.

Benutzt TextEditor.getQueryObjects() und TextEditorBean.textEditor.

String getQuerySelect ()

Definiert in Zeile 47 der Datei TextEditorBean.java.

Benutzt TextEditorBean.querySelect.

TextEditor getTextEditor ()

Definiert in Zeile 80 der Datei TextEditorBean.java.

Benutzt TextEditorBean.textEditor.

Wird benutzt von TextEditorBean.getTextItems() und TextEditorBean.setTextSelect().

List getTextItems ()

Definiert in Zeile 68 der Datei TextEditorBean.java.

Benutzt TextEditorBean.getTextEditor().

String getTextNameSelect ()

Definiert in Zeile 84 der Datei TextEditorBean.java.

Benutzt TextEditorBean.textNameSelect.

Wird benutzt von TextEditorBean.deleteText() und TextEditorBean.saveText().

String getTextSelect ()

Definiert in Zeile 33 der Datei TextEditorBean.java.

Benutzt TextEditorBean.textSelect.

Wird benutzt von TextEditorBean.saveText().

void newText ()

Definiert in Zeile 139 der Datei TextEditorBean.java.

Benutzt TextEditorBean.setTextNameSelect() und TextEditorBean.setTextSelect().

void saveText ()

Definiert in Zeile 92 der Datei TextEditorBean.java.

Benutzt TextEditorBean.getTextNameSelect() und TextEditorBean.getTextSelect().

void setQuerySelect (String querySelect)

Definiert in Zeile 51 der Datei TextEditorBean.java.

void setTextNameSelect (String textNameSelect)

Definiert in Zeile 88 der Datei TextEditorBean.java.

Wird benutzt von TextEditorBean.deleteText(), TextEditorBean.newText() und TextEditorBean.setTextSelect().

void setTextSelect (String textSelect)

Definiert in Zeile 37 der Datei TextEditorBean.java.

Benutzt TextEditorBean.getTextEditor() und TextEditorBean.setTextNameSelect().

Wird benutzt von TextEditorBean.deleteText(), TextEditorBean.newText() und TextEditorBean.textareaChange().

void textareaChange (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 132 der Datei TextEditorBean.java.

Benutzt TextEditorBean.setTextSelect().

7.3.2.4 Dokumentation der Datenelemente**String querySelect [private]**

Definiert in Zeile 28 der Datei TextEditorBean.java.

Wird benutzt von `TextEditorBean.getQuerySelect()`.

TextEditor textEditor [private]

Definiert in Zeile 27 der Datei `TextEditorBean.java`.

Wird benutzt von `TextEditorBean.getQueryList()`, `TextEditorBean.getTextEditor()` und `TextEditorBean.TextEditorBean()`.

String textNameSelect [private]

Definiert in Zeile 26 der Datei `TextEditorBean.java`.

Wird benutzt von `TextEditorBean.getTextNameSelect()`.

String textSelect [private]

Definiert in Zeile 29 der Datei `TextEditorBean.java`.

Wird benutzt von `TextEditorBean.getTextSelect()`.

7.4 Export-Implementierung

7.4.1 ContainerAdapter Klassenreferenz

Abgeleitet von `de::offis::pg::eam::mod_exp_jasper::adapter::IContainerAdapter`.

7.4.1.1 Ausführliche Beschreibung

Definiert in Zeile 9 der Datei `ContainerAdapter.java`.

Öffentliche Methoden

- `ContainerAdapter ()`
- `String getActualUser ()`
- `IContainerHandler getHandler ()`
- `ArrayList< ContainerObject > getObjects ()`
- `ArrayList< ContainerObject > getObjectsByModule (String module)`
- `ArrayList< ContainerObject > getObjectsByPrefix (String prefix)`
- `void setActualUser (String actualUser)`
- `void setHandler (IContainerHandler handler)`

Private Attribute

- String actualUser
- IContainerHandler handler

7.4.1.2 Beschreibung der Konstruktoren und Destruktoren**ContainerAdapter ()**

Definiert in Zeile 12 der Datei ContainerAdapter.java.

Benutzt ContainerAdapter.actualUser und ContainerAdapter.setHandler().

7.4.1.3 Dokumentation der Elementfunktionen**String getActualUser ()**

Implementiert IContainerAdapter.

Definiert in Zeile 62 der Datei ContainerAdapter.java.

Benutzt ContainerAdapter.actualUser.

Wird benutzt von ContainerAdapter.getObjects().

IContainerHandler getHandler ()

IContainerHandler von der Container API aus dem Kern

Implementiert IContainerAdapter.

Definiert in Zeile 52 der Datei ContainerAdapter.java.

Benutzt ContainerAdapter.handler.

Wird benutzt von ContainerAdapter.getObjects().

ArrayList<ContainerObject> getObjects ()

Definiert in Zeile 18 der Datei ContainerAdapter.java.

Benutzt ContainerAdapter.getActualUser() und ContainerAdapter.getHandler().

Wird benutzt von ContainerAdapter.getObjectsByPrefix().

ArrayList<ContainerObject> getObjectsByModule (String module)

Holt Objekte von bestimmten Modulen aus dem Container

Parameter:

module Modul Name

Rückgabe:

Elemente aus einem Modul

Implementiert IContainerAdapter.

Definiert in Zeile 45 der Datei ContainerAdapter.java.

Benutzt ContainerAdapter.getObjectsByPrefix().

ArrayList<ContainerObject> getObjectsByPrefix (String prefix)

Die Objects mit einem bestimmten Präfix werden zurückgeliefert

Rückgabe:

Liste mit Container Objects mit einem bestimmten Präfix

Implementiert IContainerAdapter.

Definiert in Zeile 28 der Datei ContainerAdapter.java.

Benutzt ContainerAdapter.getObjects().

Wird benutzt von ContainerAdapter.getObjectsByModule().

void setActualUser (String actualUser)

Definiert in Zeile 66 der Datei ContainerAdapter.java.

void setHandler (IContainerHandler handler)

Definiert in Zeile 56 der Datei ContainerAdapter.java.

Wird benutzt von ContainerAdapter.ContainerAdapter().

7.4.1.4 Dokumentation der Datenelemente

String actualUser [private]

Definiert in Zeile 11 der Datei ContainerAdapter.java.

Wird benutzt von ContainerAdapter.ContainerAdapter() und ContainerAdapter.getActualUser().

IContainerHandler handler [private]

Definiert in Zeile 10 der Datei ContainerAdapter.java.

Wird benutzt von ContainerAdapter.getHandler().

7.4.2 IContainerAdapter Schnittstellenreferenz

Basisklasse für ContainerAdapter.

7.4.2.1 Ausführliche Beschreibung

Definiert in Zeile 8 der Datei IContainerAdapter.java.

Öffentliche Methoden

- String getActualUser ()
- IContainerHandler getHandler ()
- ArrayList< ContainerObject > getObjectsByModule (String module)
- ArrayList< ContainerObject > getObjectsByPrefix (String prefix)

7.4.2.2 Dokumentation der Elementfunktionen

String getActualUser ()

Implementiert in ContainerAdapter.

IContainerHandler getHandler ()

Implementiert in ContainerAdapter.

ArrayList<ContainerObject> getObjectsByModule (String module)

Implementiert in ContainerAdapter.

Wird benutzt von Report.getMapParams().

ArrayList<ContainerObject> getObjectsByPrefix (String prefix)

Implementiert in ContainerAdapter.

Wird benutzt von Report.createData().

7.4.3 AbstractOutputFactory Klassenreferenz

Basisklasse für OutputFactory.

7.4.3.1 Ausführliche Beschreibung

Abstrakte Factory, um den Export mit Jasper Reports kompilierte Reports in ein Ausgabeformat zu ermöglichen

Autor:

Mart Koehler

Definiert in Zeile 17 der Datei AbstractOutputFactory.java.

Öffentliche Methoden

- AbstractOutputFactory (String extension, JasperPrint jasperPrint, String outputName, String outputPath)
- JRExporter getExporter ()
- String getExtension ()
- String getFilename ()
- JasperPrint getJasperPrint ()
- String getOutputName ()
- String getOutputPath ()

Geschützte Methoden

- abstract ArrayList createExporter ()

Private Attribute

- JRExporter exporter
- String extension
- String filename
- JasperPrint jasperPrint
- String outputName
- String outputPath

7.4.3.2 Beschreibung der Konstruktoren und Destruktoren

AbstractOutputFactory (String extension, JasperPrint jasperPrint, String outputName, String outputPath)

Definiert in Zeile 34 der Datei AbstractOutputFactory.java.

Benutzt AbstractOutputFactory.createExporter(), AbstractOutputFactory.exporter und AbstractOutputFactory.filename.

7.4.3.3 Dokumentation der Elementfunktionen

abstract ArrayList createExporter () [protected, pure virtual]

Die zu implementierende abstrakte Methode, in der dann die Objekte der einzelnen Datentypen erzeugt werden

Rückgabe:

vom Typ ArraList, wobei das erste Argument der komplette Dateiname und das zweite Argument das JRExporter Objekt selbst ist

Implementiert in OutputFactory.

Wird benutzt von AbstractOutputFactory.AbstractOutputFactory().

JRExporter getExporter ()

Definiert in Zeile 57 der Datei AbstractOutputFactory.java.

Benutzt AbstractOutputFactory.exporter.

Wird benutzt von Report.exportCompiledDesign().

String getExtension ()

Definiert in Zeile 53 der Datei AbstractOutputFactory.java.

Benutzt AbstractOutputFactory.extension.

Wird benutzt von OutputFactory.createExporter().

String getFilename ()

Definiert in Zeile 65 der Datei AbstractOutputFactory.java.

Benutzt AbstractOutputFactory.filename.

Wird benutzt von Report.exportCompiledDesign().

JasperPrint getJasperPrint ()

Definiert in Zeile 30 der Datei AbstractOutputFactory.java.

Benutzt AbstractOutputFactory.jasperPrint.

Wird benutzt von OutputFactory.createExporter().

String getOutputName ()

Definiert in Zeile 26 der Datei AbstractOutputFactory.java.

Benutzt AbstractOutputFactory.outputName.

Wird benutzt von OutputFactory.createExporter().

String getOutputPath ()

Definiert in Zeile 61 der Datei `AbstractOutputFactory.java`.

Benutzt `AbstractOutputFactory.outputPath`.

Wird benutzt von `OutputFactory.createExporter()`.

7.4.3.4 Dokumentation der Datenelemente

JRExporter exporter [private]

Definiert in Zeile 18 der Datei `AbstractOutputFactory.java`.

Wird benutzt von `AbstractOutputFactory.AbstractOutputFactory()` und `AbstractOutputFactory.getExporter()`.

String extension [private]

Definiert in Zeile 19 der Datei `AbstractOutputFactory.java`.

Wird benutzt von `AbstractOutputFactory.getExtension()`.

String filename [private]

Definiert in Zeile 23 der Datei `AbstractOutputFactory.java`.

Wird benutzt von `AbstractOutputFactory.AbstractOutputFactory()` und `AbstractOutputFactory.getFilename()`.

JasperPrint jasperPrint [private]

Definiert in Zeile 22 der Datei `AbstractOutputFactory.java`.

Wird benutzt von `AbstractOutputFactory.getJasperPrint()`.

String outputName [private]

Definiert in Zeile 20 der Datei `AbstractOutputFactory.java`.

Wird benutzt von `AbstractOutputFactory.getOutputName()`.

String outputPath [private]

Definiert in Zeile 21 der Datei `AbstractOutputFactory.java`.

Wird benutzt von `AbstractOutputFactory.getOutputPath()`.

7.4.4 Context Klassenreferenz

7.4.4.1 Ausführliche Beschreibung

Benötigt, um den Pfad der config.xml zu bekommen

Definiert in Zeile 15 der Datei Context.java.

Öffentliche, statische Methoden

- static BundleContext getContext ()
- static void setContext (BundleContext context)

Statische private Attribute

- static BundleContext context
- static String path

7.4.4.2 Dokumentation der Elementfunktionen

static BundleContext getContext () [static]

Definiert in Zeile 22 der Datei Context.java.

Benutzt Context.context.

static void setContext (BundleContext context) [static]

Definiert in Zeile 27 der Datei Context.java.

7.4.4.3 Dokumentation der Datenelemente

BundleContext context [static, private]

Definiert in Zeile 18 der Datei Context.java.

Wird benutzt von Context.getContext().

String path [static, private]

Definiert in Zeile 19 der Datei Context.java.

7.4.5 CsvExporter Klassenreferenz

Abgeleitet von net::sf::jasperreports::engine::export::JRCsvExporter und de::offis::pg::eam::mod_exp_jasper::IExporter.

7.4.5.1 Ausführliche Beschreibung

Definiert in Zeile 13 der Datei CsvExporter.java.

Öffentliche Methoden

- CsvExporter (JasperPrint jasperPrint, String output)

7.4.5.2 Beschreibung der Konstruktoren und Destruktoren

CsvExporter (JasperPrint jasperPrint, String output)

Definiert in Zeile 15 der Datei CsvExporter.java.

7.4.6 HtmlExporter Klassenreferenz

Abgeleitet von `net::sf::jasperreports::engine::export::JRHtmlExporter` und `de::offis::pg::eam::mod_exp_jasper::IExporter`.

7.4.6.1 Ausführliche Beschreibung

Implementierung der Verarbeitung zu HTML

Autor:

Mart Koehler

Definiert in Zeile 20 der Datei HtmlExporter.java.

Öffentliche Methoden

- HtmlExporter (JasperPrint jasperPrint, String output)

7.4.6.2 Beschreibung der Konstruktoren und Destruktoren

HtmlExporter (JasperPrint jasperPrint, String output)

Definiert in Zeile 22 der Datei HtmlExporter.java.

7.4.7 RtfExporter Klassenreferenz

Abgeleitet von `net::sf::jasperreports::engine::export::JRRtfExporter` und `de::offis::pg::eam::mod_exp_jasper::IExporter`.

7.4.7.1 Ausführliche Beschreibung

Definiert in Zeile 11 der Datei RtfExporter.java.

Öffentliche Methoden

- RtfExporter (JasperPrint jasperPrint, String output)

7.4.7.2 Beschreibung der Konstruktoren und Destruktoren

RtfExporter (JasperPrint jasperPrint, String output)

Definiert in Zeile 12 der Datei RtfExporter.java.

7.4.8 TxtExporter Klassenreferenz

Abgeleitet von net::sf::jasperreports::engine::export::JRTextExporter und de::offis::pg::eam::mod_exp_jasper::IExporter.

7.4.8.1 Ausführliche Beschreibung

Definiert in Zeile 12 der Datei TxtExporter.java.

Öffentliche Methoden

- TxtExporter (JasperPrint jasperPrint, String output)

7.4.8.2 Beschreibung der Konstruktoren und Destruktoren

TxtExporter (JasperPrint jasperPrint, String output)

Definiert in Zeile 13 der Datei TxtExporter.java.

7.4.9 XlsExporter Klassenreferenz

Abgeleitet von net::sf::jasperreports::engine::export::JRXlsExporter und de::offis::pg::eam::mod_exp_jasper::IExporter.

7.4.9.1 Ausführliche Beschreibung

Der Export in Excel wird über diese Klasse abgewickelt, da Jasper Reports keine einfache Methode anbietet.

Autor:

Mart Koehler

Definiert in Zeile 17 der Datei XlsExporter.java.

Öffentliche Methoden

- XlsExporter (JasperPrint jasperPrint, String output)

7.4.9.2 Beschreibung der Konstruktoren und Destruktoren

XlsExporter (JasperPrint jasperPrint, String output)

Definiert in Zeile 18 der Datei XlsExporter.java.

7.4.10 XmlExporter Klassenreferenz

Abgeleitet von `net::sf::jasperreports::engine::export::JRXmlExporter` und `de::offis::pg::eam::mod_exp_jasper::IExporter`.

7.4.10.1 Ausführliche Beschreibung

Definiert in Zeile 12 der Datei XmlExporter.java.

Öffentliche Methoden

- XmlExporter (JasperPrint jasperPrint, String output)

7.4.10.2 Beschreibung der Konstruktoren und Destruktoren

XmlExporter (JasperPrint jasperPrint, String output)

Definiert in Zeile 13 der Datei XmlExporter.java.

7.4.11 IExporter Schnittstellenreferenz

Basisklasse für `CsvExporter`, `HtmlExporter`, `RtfExporter`, `TxtExporter`, `XlsExporter` und `XmlExporter`.

7.4.11.1 Ausführliche Beschreibung

Definiert in Zeile 5 der Datei IExporter.java.

7.4.12 Init Klassenreferenz

7.4.12.1 Ausführliche Beschreibung

Definiert in Zeile 8 der Datei Init.java.

Öffentliche Methoden

- JasperPrint getJasperPrint ()
- JasperReport getJasperReport ()
- HashMap< String, String > getParameter ()
- void setJasperPrint (JasperPrint jasperPrint)
- void setJasperReport (JasperReport jasperReport)
- void setParameter (HashMap< String, String > parameter)

Private Attribute

- JasperPrint jasperPrint
- JasperReport jasperReport
- HashMap< String, String > parameter

7.4.12.2 Dokumentation der Elementfunktionen

JasperPrint getJasperPrint ()

Definiert in Zeile 20 der Datei Init.java.

Benutzt Init.jasperPrint.

JasperReport getJasperReport ()

Definiert in Zeile 14 der Datei Init.java.

Benutzt Init.jasperReport.

Wird benutzt von Report.createDesignObject().

HashMap<String, String> getParameter ()

Definiert in Zeile 26 der Datei Init.java.

Benutzt Init.parameter.

void setJasperPrint (JasperPrint jasperPrint)

Definiert in Zeile 23 der Datei Init.java.

Wird benutzt von `Report.fillDesign()`.

`void setJasperReport (JasperReport jasperReport)`

Definiert in Zeile 17 der Datei `Init.java`.

Wird benutzt von `Report.compileDesign()`.

`void setParameter (HashMap< String, String > parameter)`

Definiert in Zeile 29 der Datei `Init.java`.

7.4.12.3 Dokumentation der Datenelemente

`JasperPrint jasperPrint` [private]

Definiert in Zeile 10 der Datei `Init.java`.

Wird benutzt von `Init.getJasperPrint()`.

`JasperReport jasperReport` [private]

Definiert in Zeile 9 der Datei `Init.java`.

Wird benutzt von `Init.getJasperReport()`.

`HashMap<String, String> parameter` [private]

Initialisierung:

```
new HashMap<String, String>()
```

Definiert in Zeile 11 der Datei `Init.java`.

Wird benutzt von `Init.getParameter()`.

7.4.13 OutputFactory Klassenreferenz

Abgeleitet von `de::offis::pg::eam::mod_exp_jasper::AbstractOutputFactory`.

7.4.13.1 Ausführliche Beschreibung

Erweiterung zur `AbstractOutputFactory` Weitere Datentypen können über eine entsprechende Klasse im Package `de.offis.pg.eam.mod_exp_jasper.extension` und und der Methode `createExporter` eingebunden werden

Autor:

Mart Koehler

Definiert in Zeile 32 der Datei OutputFactory.java.

Öffentliche Methoden

- Calendar getDate ()
- OutputFactory (String extension, JasperPrint jasperPrint, String outputName, String outputPath)

Geschützte Methoden

- ArrayList createExporter ()

Private Methoden

- void setDate (Calendar date)

Private Attribute

- Calendar date

7.4.13.2 Beschreibung der Konstruktoren und Destruktoren

OutputFactory (String extension, JasperPrint jasperPrint, String outputName, String outputPath)

Definiert in Zeile 35 der Datei OutputFactory.java.

7.4.13.3 Dokumentation der Elementfunktionen

ArrayList createExporter () [protected, virtual]

Implementiert AbstractOutputFactory.

Definiert in Zeile 42 der Datei OutputFactory.java.

Benutzt AbstractOutputFactory.getExtension(), AbstractOutputFactory.getJasperPrint(), AbstractOutputFactory.getOutputName() und AbstractOutputFactory.getOutputPath().

Calendar getDate ()

Definiert in Zeile 85 der Datei OutputFactory.java.

Benutzt OutputFactory.date und OutputFactory.setDate().

void setDate (Calendar date) [private]

Definiert in Zeile 92 der Datei OutputFactory.java.

Wird benutzt von OutputFactory.getDate().

7.4.13.4 Dokumentation der Datenelemente

Calendar date [private]

Definiert in Zeile 34 der Datei OutputFactory.java.

Wird benutzt von OutputFactory.getDate().

7.4.14 Report Klassenreferenz

Basisklasse für ReportFrontend.

7.4.14.1 Ausführliche Beschreibung

Standardmethoden, die für alle Designs gelten, werden für den Export hier bereitgestellt.

Autor:

Mart Koehler

Definiert in Zeile 35 der Datei Report.java.

Öffentliche Methoden

- void compileDesign ()
- JRMapCollectionDataSource createData ()
- String createDesignObject ()
- String exportCompiledDesign ()
- ArrayList< String > fileListing (String path)
- ArrayList< String > fileOutputListing (String path)
- boolean fillDesign ()
- String getActualUser ()
- IContainerAdapter getAdapter ()
- Design getDesign ()
- String getDesignFilename ()
- String getDesignPath ()
- Init getInit ()
- HashMap< String, Object > getMapParams ()
- String getOutputName ()

- String getOutputPath ()
- String getOutputType ()
- ArrayList< String > getSelFields ()
- TextEditor getTextEditor ()
- Report ()
- void setActualUser (String actualUser)
- void setDesign (Design design)
- void setDesignFilename (String designFilename)
- void setDesignPath (String designPath)
- void setInit (Init init)
- void setMapParams (HashMap< String, Object > mapParams)
- void setOutputName (String outputName)
- void setOutputPath (String outputPath)
- void setOutputType (String outputType)
- void setSelFields (ArrayList< String > selFields)
- void setTextEditor (TextEditor textEditor)

Private Attribute

- String actualUser
- IContainerAdapter adapter
- Design design
- String designFilename
- String designPath
- Init init
- HashMap< String, Object > mapParams
- String outputName
- String outputPath
- String outputType
- ArrayList< String > selFields
- TextEditor textEditor

7.4.14.2 Beschreibung der Konstruktoren und Destruktoren

Report ()

Definiert in Zeile 49 der Datei Report.java.

Benutzt Report.adapter, Report.designFilename, Report.init, Report.mapParams, Report.outputName, Report.setDesignPath(), Report.setOutputPath() und Report.textEditor.

7.4.14.3 Dokumentation der Elementfunktionen

void compileDesign ()

Die XML Design Datei wird hier kompiliert

Definiert in Zeile 96 der Datei Report.java.

Benutzt Report.fileListing(), Report.getDesignFilename(), Report.getDesignPath(), Report.getInit() und Init.setJasperReport().

Wird benutzt von Report.createDesignObject().

JRMapCollectionDataSource createData ()

Hier werden die nötigen Daten vom Container und Textgenerator geholt und anhand der Parameter, Felder und Variablen zusammen gebaut

Rückgabe:

Das Ergebnis wird als JRMapCollectionDataSource verpackt und an den Report gegeben

Definiert in Zeile 133 der Datei Report.java.

Benutzt TextEditor.buildData(), Report.getAdapter(), Report.getDesign(), Design.getFields(), IContainerAdapter.getObjectsByPrefix(), Report.getSelFields() und Report.getTextEditor().

Wird benutzt von Report.fillDesign().

String createDesignObject ()

Design Objekt im System wird erzeugt, um Manipulationen zuzulassen

Rückgabe:

Rückgabe für JSF Site Forwarding interessant

Definiert in Zeile 249 der Datei Report.java.

Benutzt Report.compileDesign(), Report.getDesign(), Report.getDesignFilename(), Report.getInit(), Init.getJasperReport(), Report.setDesign(), Design.setFields(), Design.setFilename(), Design.setParameters() und Design.setVars().

Wird benutzt von ReportFrontend.createDesign().

String exportCompiledDesign ()

Abschließendes exportieren des Reports in ein Ausgabeformat

Rückgabe:

Für die Page Navigation in JSF wichtig. Ausgabe entweder "success" oder "failure"

Definiert in Zeile 188 der Datei Report.java.

Benutzt Report.fillDesign(), AbstractOutputFactory.getExporter(), AbstractOutputFactory.getFilename(), Report.getInit(), Report.getOutputName(), Report.getOutputPath() und Report.getOutputType().

Wird benutzt von ReportFrontend.exportDesign().

ArrayList<String> fileListing (String path)

Sucht aus einem Verzeichnis alle Design Dateien

Rückgabe:

Liefert eine ArrayList<String> der Namen zurück

Definiert in Zeile 280 der Datei Report.java.

Benutzt Report.getDesignPath() und Report.getOutputPath().

Wird benutzt von Report.compileDesign() und ReportFrontend.getDesignFileItems().

ArrayList<String> fileOutputListing (String path)

Definiert in Zeile 302 der Datei Report.java.

Wird benutzt von ReportFrontend.getOutputFiles().

boolean fillDesign ()

Das Design mit den Parametern füllen

Definiert in Zeile 115 der Datei Report.java.

Benutzt Report.createData(), Report.getInit(), Report.getMapParams() und Init.setJasperPrint().

Wird benutzt von Report.exportCompiledDesign().

String getActualUser ()

Definiert in Zeile 434 der Datei Report.java.

Benutzt Report.actualUser und Report.setActualUser().

Wird benutzt von ReportFrontend.DbDocMerge() und ReportFrontend.getOutputFiles().

IContainerAdapter getAdapter ()

Definiert in Zeile 430 der Datei Report.java.

Benutzt Report.adapter.

Wird benutzt von Report.createData() und Report.getMapParams().

Design getDesign ()

Definiert in Zeile 368 der Datei Report.java.

Benutzt Report.design.

Wird benutzt von Report.createData(), Report.createDesignObject() und ReportFrontend.getFieldItems().

String getDesignFilename ()

Definiert in Zeile 392 der Datei Report.java.

Benutzt Report.designFilename.

Wird benutzt von Report.compileDesign(), Report.createDesignObject(), ReportFrontend.designFileValueListener() und ReportFrontend.exportDesign().

String getDesignPath ()

Definiert in Zeile 328 der Datei Report.java.

Benutzt Report.designPath.

Wird benutzt von Report.compileDesign(), Report.fileListing(), ReportFrontend.getDesignFileItems(), Report.getMapParams() und Report.setDesignPath().

Init getInit ()

Definiert in Zeile 351 der Datei Report.java.

Benutzt Report.init.

Wird benutzt von Report.compileDesign(), Report.createDesignObject(), Report.exportCompiledDesign() und Report.fillDesign().

HashMap<String, Object> getMapParams ()

getMapParams heißt in diesem Falle die map mit Daten aus dem Container füllen und zurückzugeben Hierüber werden auch Bilder aus dem Container geholt "imagePath_n" n Element natürlichen Zahlen ohne 0

Rückgabe:

Parameter Liste

Definiert in Zeile 448 der Datei Report.java.

Benutzt Report.getAdapter(), Report.getDesignPath(), IContainerAdapter.getObjectsByModule() und Report.mapParams.

Wird benutzt von Report.fillDesign().

String getOutputName ()

Definiert in Zeile 358 der Datei Report.java.

Benutzt Report.outputName.

Wird benutzt von Report.exportCompiledDesign() und Report.setOutputName().

String getOutputPath ()

Definiert in Zeile 400 der Datei Report.java.

Benutzt Report.outputPath.

Wird benutzt von ReportFrontend.DbDocMerge(), ReportFrontend.download(), Report.exportCompiledDesign(), Report.fileListing(), ReportFrontend.getOutputFiles() und Report.setOutputPath().

String getOutputType ()

Definiert in Zeile 320 der Datei Report.java.

Benutzt Report.outputType.

Wird benutzt von Report.exportCompiledDesign() und Report.setOutputType().

ArrayList<String> getSelFields ()

Definiert in Zeile 382 der Datei Report.java.

Benutzt Report.selFields.

Wird benutzt von Report.createData().

TextEditor getTextEditor ()

Definiert in Zeile 85 der Datei Report.java.

Benutzt Report.textEditor.

Wird benutzt von Report.createData().

void setActualUser (String actualUser)

Definiert in Zeile 439 der Datei Report.java.

Wird benutzt von Report.getActualUser() und Report.setOutputPath().

void setDesign (Design design)

Definiert in Zeile 377 der Datei Report.java.

Wird benutzt von Report.createDesignObject().

void setDesignFilename (String designFilename)

Definiert in Zeile 396 der Datei Report.java.

Wird benutzt von ReportFrontend.designFileValueListener().

void setDesignPath (String designPath)

Definiert in Zeile 331 der Datei Report.java.

Benutzt Report.getDesignPath().

Wird benutzt von Report.Report().

void setInit (Init init)

Definiert in Zeile 354 der Datei Report.java.

void setMapParams (HashMap< String, Object > mapParams)

Definiert in Zeile 468 der Datei Report.java.

void setOutputName (String outputName)

Definiert in Zeile 362 der Datei Report.java.

Benutzt Report.getOutputName().

void setOutputPath (String outputPath)

Definiert in Zeile 404 der Datei Report.java.

Benutzt Report.getOutputPath() und Report.setActualUser().

Wird benutzt von Report.Report().

void setOutputType (String outputType)

Definiert in Zeile 323 der Datei Report.java.

Benutzt Report.getOutputType().

void setSelFields (ArrayList< String > selFields)

Definiert in Zeile 386 der Datei Report.java.

void setTextEditor (TextEditor textEditor)

Definiert in Zeile 89 der Datei Report.java.

7.4.14.4 Dokumentation der Datenelemente

String actualUser [private]

Definiert in Zeile 48 der Datei Report.java.

Wird benutzt von Report.getActualUser().

IContainerAdapter adapter [private]

Definiert in Zeile 47 der Datei Report.java.

Wird benutzt von Report.getAdapter() und Report.Report().

Design design [private]

Definiert in Zeile 44 der Datei Report.java.

Wird benutzt von Report.getDesign().

String designFilename [private]

Definiert in Zeile 40 der Datei Report.java.

Wird benutzt von Report.getDesignFilename() und Report.Report().

String designPath [private]

Definiert in Zeile 41 der Datei Report.java.

Wird benutzt von Report.getDesignPath().

Init init [private]

Definiert in Zeile 46 der Datei Report.java.

Wird benutzt von Report.getInit() und Report.Report().

HashMap<String, Object> mapParams [private]

Definiert in Zeile 37 der Datei Report.java.

Wird benutzt von Report.getMapParams() und Report.Report().

String outputName [private]

Definiert in Zeile 43 der Datei Report.java.

Wird benutzt von Report.getOutputName() und Report.Report().

String outputPath [private]

Definiert in Zeile 42 der Datei Report.java.

Wird benutzt von Report.getOutputPath().

String outputType [private]

Definiert in Zeile 38 der Datei Report.java.

Wird benutzt von Report.getOutputType().

ArrayList<String> selFields [private]

Definiert in Zeile 45 der Datei Report.java.

Wird benutzt von Report.getSelFields().

TextEditor textEditor [private]

Definiert in Zeile 36 der Datei Report.java.

Wird benutzt von Report.getTextEditor() und Report.Report().

7.4.15 ReportFrontend Klassenreferenz

Abgeleitet von *de::offis::pg::eam::mod_exp_jasper::Report*.

7.4.15.1 Ausführliche Beschreibung

Für die Übersicht werden die Methoden für die Präsentation getrennt, indem die Logik Klasse Report erweitert wird.

Hier finden sich alle Methoden, die von JSF angesprochen werden, um das Formular aufzubauen

Autor:

Mart Koehler

Definiert in Zeile 30 der Datei ReportFrontend.java.

Öffentliche Methoden

- String confirm ()
- String createDesign ()
- void designFileValueListener (ValueChangeEvent valueChangeEvent)
- void download ()
- String exportDesign ()

- String getCompiledExportsCount ()
- List getDesignFileItems ()
- List getExporterItems ()
- List getFieldItems ()
- javax.faces.component.html.HtmlInputHidden getHiddenDownload ()
- List getOutputFiles ()
- ReportFrontend ()
- void setHiddenDownload (javax.faces.component.html.HtmlInputHidden hiddenDownload)

Private Methoden

- void DbDocMerge ()

Private Attribute

- javax.faces.component.html.HtmlInputHidden hiddenDownload

7.4.15.2 Beschreibung der Konstruktoren und Destruktoren

ReportFrontend ()

Definiert in Zeile 43 der Datei ReportFrontend.java.

7.4.15.3 Dokumentation der Elementfunktionen

String confirm ()

Definiert in Zeile 207 der Datei ReportFrontend.java.

String createDesign ()

Delegation

Rückgabe:

Definiert in Zeile 215 der Datei ReportFrontend.java.

Benutzt Report.createDesignObject().

Wird benutzt von ReportFrontend.designFileValueListener().

void DbDocMerge () [private]

Verwaltete Dokumente werden auf Existenz geprüft und gelöscht, falls das Dokument nicht mehr existiert

Definiert in Zeile 110 der Datei ReportFrontend.java.

Benutzt Report.getActualUser() und Report.getOutputPath().

Wird benutzt von ReportFrontend.getOutputFiles().

void designFileValueListener (ValueChangeEvent valueChangeEvent)

Listener für die Auswahlbox in JSF

Parameter:

valueChangeEvent

Definiert in Zeile 158 der Datei ReportFrontend.java.

Benutzt ReportFrontend.createDesign(), Report.getDesignFilename() und Report.setDesignFilename().

void download ()

Download Servlet Implementierung

Definiert in Zeile 239 der Datei ReportFrontend.java.

Benutzt ReportFrontend.getHiddenDownload() und Report.getOutputPath().

String exportDesign ()

Delegation

Rückgabe:

Message für die Navigation in JSF

Definiert in Zeile 223 der Datei ReportFrontend.java.

Benutzt Report.exportCompiledDesign() und Report.getDesignFilename().

String getCompiledExportsCount ()

Fragt die Anzahl bisher getätigter Reportgenerierungen durch

Rückgabe:

genaue Anzahl wieviele Reports von diesem User erzeugt wurden

Definiert in Zeile 285 der Datei ReportFrontend.java.

List getDesignFileItems ()

Alle Dateien aus dem Design Ordner in eine Liste aufnehmen

Rückgabe:

Die Dateien als Liste zurückgeben

Definiert in Zeile 51 der Datei ReportFrontend.java.

Benutzt Report.listFiles() und Report.getDesignPath().

List getExporterItems ()

Verfügbare Extensions aus der Datenbank ziehen

Rückgabe:

Definiert in Zeile 172 der Datei ReportFrontend.java.

List getFieldItems ()

Felder aus einem Design Template auslesen

Rückgabe:

Felder als Liste zurückgeben

Definiert in Zeile 144 der Datei ReportFrontend.java.

Benutzt Report.getDesign() und Design.getFields().

javax.faces.component.html.HtmlInputHidden getHiddenDownload ()

Definiert in Zeile 34 der Datei ReportFrontend.java.

Benutzt ReportFrontend.hiddenDownload.

Wird benutzt von ReportFrontend.download().

List getOutputFiles ()

Liste der möglichen zu runterladenden Files

Rückgabe:

Definiert in Zeile 71 der Datei ReportFrontend.java.

Benutzt ReportFrontend.DbDocMerge(), Report.fileOutputListing(),

Report.getActualUser() und Report.getOutputPath().

void setHiddenDownload (javax.faces.component.html.HtmlInputHidden hiddenDownload)

Definiert in Zeile 38 der Datei ReportFrontend.java.

Benutzt ReportFrontend.hiddenDownload.

7.4.15.4 Dokumentation der Datenelemente

javax.faces.component.html.HtmlInputHidden hiddenDownload [private]

Definiert in Zeile 32 der Datei ReportFrontend.java.

Wird benutzt von ReportFrontend.getHiddenDownload() und ReportFrontend.setHiddenDownload().

7.4.16 ReportTest Klassenreferenz

7.4.16.1 Ausführliche Beschreibung

Testklasse

Autor:

Mart Koehler

Definiert in Zeile 14 der Datei ReportTest.java.

Öffentliche, statische Methoden

- static void main (String[] args)

7.4.16.2 Dokumentation der Elementfunktionen

static void main (String[] args) [static]

Parameter:

args

IN JSF -Muss ein Ausgabeformat gewählt werden -Template gewählt werden -Parameter gesetzt werden -Ausgabeort wählen -compile anfordern

setOutput geht dann über eine DropDown Liste

Definiert in Zeile 19 der Datei ReportTest.java.

7.4.17 MessageTag Klassenreferenz

Abgeleitet von javax::servlet::jsp::tagext::TagSupport.

7.4.17.1 Ausführliche Beschreibung

Diese Klasse ist fuer die Taglibrary des Sicherheitssystem. Mit Hilfe dieses Tags werden Bereiche auf einer Seite mit einer Rolle/ID/Recht gekennzeichnet. Diese/s Rolle/ID/Recht wird spaeter von Admin dann auf Rollen gemappt die existieren.

Autor:

christian zillmann

Definiert in Zeile 22 der Datei MessageTag.java.

Öffentliche Methoden

- `int doAfterBody ()` throws `JspException`
- `int doEndTag ()` throws `JspException`
- `int doStartTag ()` throws `JspException`
- `void setMessageAttribut (String messageAttribut)`
- `void setPageContext (PageContext pageContext)`

Private Attribute

- `String messageAttribut`

Statische private Attribute

- `static final long serialVersionUID = 1L`

7.4.17.2 Dokumentation der Elementfunktionen

`int doAfterBody ()` throws `JspException`

Definiert in Zeile 43 der Datei MessageTag.java.

`int doEndTag ()` throws `JspException`

Definiert in Zeile 54 der Datei MessageTag.java.

`int doStartTag ()` throws `JspException`

Definiert in Zeile 63 der Datei MessageTag.java.

Benutzt `MessageTag.messageAttribut`.

void setMessageAttribut (String messageAttribut)

Definiert in Zeile 33 der Datei MessageTag.java.

void setPageContext (PageContext pageContext)

Definiert in Zeile 92 der Datei MessageTag.java.

7.4.17.3 Dokumentation der Datenelemente

String messageAttribut [private]

Definiert in Zeile 29 der Datei MessageTag.java.

Wird benutzt von MessageTag.doStartTag().

final long serialVersionUID = 1L [static, private]

Definiert in Zeile 27 der Datei MessageTag.java.

8 de.offis.pg.eam.mod_analysis

8.1 BundleConfiguration

8.1.1 BundleViews Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::module::api::IModuleView.

8.1.1.1 Ausführliche Beschreibung

Definiert in Zeile 22 der Datei BundleViews.java.

Öffentliche Methoden

- String getBundleVersion ()
- String getDescription ()
- long getEquinoxBundleId ()
- String getName ()
- List< View > getViews ()

Statische private Attribute

- static final String VIEWS = "bundle_views.xml"

8.1.1.2 Dokumentation der Elementfunktionen

String getBundleVersion ()

Definiert in Zeile 84 der Datei BundleViews.java.

String getDescription ()

Definiert in Zeile 29 der Datei BundleViews.java.

long getEquinoxBundleId ()

Definiert in Zeile 35 der Datei BundleViews.java.

String getName ()

Definiert in Zeile 41 der Datei BundleViews.java.

List<View> getViews ()

Definiert in Zeile 47 der Datei BundleViews.java.

Benutzt BundleViews.VIEWS.

8.1.1.3 Dokumentation der Datenelemente

final String VIEWS = "bundle_views.xml" [static, private]

Definiert in Zeile 25 der Datei BundleViews.java.

Wird benutzt von BundleViews.getViews().

8.1.2 Configuration Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::configuration::api::IBundleConfiguration`.

8.1.2.1 Ausführliche Beschreibung

Definiert in Zeile 10 der Datei Configuration.java.

Öffentliche Methoden

- `boolean bundleHasObjects ()`
- `boolean bundleHasRelations ()`
- `boolean bundleHasViews ()`
- `Document getObjectsJdomDocument ()` throws `JDOMException`, `IOException`

8.1.2.2 Dokumentation der Elementfunktionen

boolean bundleHasObjects ()

Definiert in Zeile 13 der Datei Configuration.java.

boolean bundleHasRelations ()

Definiert in Zeile 28 der Datei Configuration.java.

boolean bundleHasViews ()

Definiert in Zeile 18 der Datei Configuration.java.

Document getObjectJdomDocument () throws JDOMException, IOException

Definiert in Zeile 23 der Datei Configuration.java.

8.1.3 Activator Klassenreferenz

Abgeleitet von org::osgi::framework::BundleActivator.

8.1.3.1 Ausführliche Beschreibung

Definiert in Zeile 6 der Datei Activator.java.

Öffentliche Methoden

- void start (BundleContext context) throws Exception
- void stop (BundleContext context) throws Exception

Statische öffentliche Attribute

- static String bundleSymbolicName
- static String bundleVersion
- static long equinoxBundleId

8.1.3.2 Dokumentation der Elementfunktionen**void start (BundleContext context) throws Exception**

Definiert in Zeile 16 der Datei Activator.java.

Benutzt Activator.bundleSymbolicName, Activator.bundleVersion und Activator.equinoxBundleId.

void stop (BundleContext context) throws Exception

Definiert in Zeile 28 der Datei Activator.java.

8.1.3.3 Dokumentation der Datenelemente**String bundleSymbolicName [static]**

Definiert in Zeile 8 der Datei Activator.java.

Wird benutzt von Activator.start().

String bundleVersion [static]

Definiert in Zeile 9 der Datei Activator.java.

Wird benutzt von Activator.start().

long equinoxBundleId [static]

Definiert in Zeile 10 der Datei Activator.java.

Wird benutzt von Activator.start().

8.2 API

8.2.1 AnalysisFactory Klassenreferenz

8.2.1.1 Ausführliche Beschreibung

Definiert in Zeile 11 der Datei AnalysisFactory.java.

Öffentliche Methoden

- AnalysisFactory (HashMap< String, String > parameter, int option, ArrayList< DataObject > obj, String user, String passwd)
- IDataCollector getCollector ()

Paketattribute

- IDataCollector collector

8.2.1.2 Beschreibung der Konstruktoren und Destruktoren

AnalysisFactory (HashMap< String, String > parameter, int option, ArrayList< DataObject > obj, String user, String passwd)

Definiert in Zeile 13 der Datei AnalysisFactory.java.

Benutzt AnalysisFactory.collector.

8.2.1.3 Dokumentation der Elementfunktionen

IDataCollector getCollector ()

Definiert in Zeile 18 der Datei AnalysisFactory.java.

Benutzt AnalysisFactory.collector.

Wird benutzt von `Test.main()`.

8.2.1.4 Dokumentation der Datenelemente

IDataCollector collector [package]

Definiert in Zeile 12 der Datei `AnalysisFactory.java`.

Wird benutzt von `AnalysisFactory.AnalysisFactory()` und `AnalysisFactory.getCollector()`.

8.2.2 IDataCollector Schnittstellenreferenz

Basisklasse für `DataCollector`.

8.2.2.1 Ausführliche Beschreibung

Definiert in Zeile 8 der Datei `IDataCollector.java`.

Öffentliche Methoden

- `boolean findInTree (DataObject obj, TreeObject tree)`
- `ArrayList< TreeObject > getCompleteTree ()`
- `int getDepth (TreeObject tree)`
- `ArrayList< DataObject > getElementsFromLevel (int level)`
- `ArrayList< DataObject > getObjectsFromLevel (int level)`
- `ArrayList< DataObject > getOccurrence ()`
- `String getPerson ()`
- `String getRelationPerson ()`
- `ArrayList< DataObject > getRelationsFromLevel (int level)`
- `ArrayList< TreeObject > getRelationTreesFromLevel (ArrayList< TreeObject > treeElem, int level)`
- `TreeObject getTree (int index)`
- `XmlConverter getXMLConverter ()`
- `void setPerson (String person)`
- `void setRelationPerson (String relationPerson)`

8.2.2.2 Dokumentation der Elementfunktionen

`boolean findInTree (DataObject obj, TreeObject tree)`

Findet ein bestimmtes Object in einem Baum

Parameter:

obj Das gesuchte Objekt

tree In diesem Baum soll gesucht werden

Rückgabe:

Objekt gefunden oder nicht

Implementiert in DataCollector.

Wird benutzt von Test.main().

ArrayList<TreeObject> getCompleteTree ()

Gibt alle Bäume zu allen zuvor übergebenen Objekten komplett als ArrayList zurück

Rückgabe:

ArrayList<TreeObject>

Implementiert in DataCollector.

Wird benutzt von Test.main().

int getDepth (TreeObject tree)

Die maximale Tiefe eines Baumes

Parameter:

tree der zu untersuchende Baum

Rückgabe:

die maximale Tiefe

Implementiert in DataCollector.

Wird benutzt von Test.main() und XmlConverter.XmlConverter().

ArrayList<DataObject> getElementsFromLevel (int level)

Holt alle Objekte eines bestimmten Levels

Parameter:

level Stufe, die zurückgegeben werden soll

Rückgabe:

Liste von DataObjects

Implementiert in DataCollector.

ArrayList<DataObject> getObjectsFromLevel (int level)

Holt eine Liste von DataObjects, die als MetaObject ein EAMObject sind

Parameter:

level Stufe, die zurückgegeben werden soll

Rückgabe:

Liste von DataObjects

Implementiert in DataCollector.

Wird benutzt von XmlConverter.XmlConverter().

ArrayList<DataObject> getOccurrence ()

Die doppelten Objekte werden hier ausgegeben.

Rückgabe:

Entweder die List an doppelten DataObjects oder null

Implementiert in DataCollector.

String getPerson ()

Name des Objektes, welches man gesetzt hat. Default: verantwortlicher

Rückgabe:

Name des Objektes

Implementiert in DataCollector.

String getRelationPerson ()

Name der Relation, die ein Objekt mit dem Personen objekt verknüpft. Default: hat

Rückgabe:

Name der Relation, die man gesetzt hat

Implementiert in DataCollector.

ArrayList<DataObject> getRelationsFromLevel (int level)

Holt eine Liste von DataObjects, die als MetaObject ein EAMRelation sind

Parameter:

level Stufe, die zurückgegeben werden soll

Rückgabe:

Liste von DataObjects

Implementiert in DataCollector.

ArrayList<TreeObject> getRelationTreesFromLevel (ArrayList< TreeObject > treeElem, int level)

Holt Relationsobjekte eingekapselt in einem TreeObjekt

Parameter:

level Stufe, die zurückgegeben werden soll

Rückgabe:

Liste von TreeObjekts mit Relationen

Implementiert in DataCollector.

Wird benutzt von XmlConverter.XmlConverter().

TreeObject getTree (int index)

Holt mir einen bestimmten Baum aus mehreren Bäumen

Parameter:

index Der Index des Array Elements. Die Gesamtlänge kann mit getCompleteTree ermittelt werden

Rückgabe:

ein TreeObject aus der Liste

Implementiert in DataCollector.

Wird benutzt von Test.main().

XmlConverter getXMLConverter ()

Konvertiert den Baum nach XML für die spezielle SVG Grafiken

Rückgabe:

der Converter selbst

Implementiert in DataCollector.

void setPerson (String person)

Name eines Objektes, den man für die Analyse einsetzen möchte. Dabei muss das Objekt die Attribute Name, Vorname, Telefon besitzen

Parameter:

person Name für das Objekt

Implementiert in DataCollector.

void setRelationPerson (String relationPerson)

Setzen des Relationennamens bzgl eines Personen Objekts

Parameter:

relationPerson Name als String

Implementiert in DataCollector.

8.2.3 IXmlConverter Schnittstellenreferenz

Basisklasse für XmlConverter.

8.2.3.1 Ausführliche Beschreibung

Definiert in Zeile 8 der Datei IXmlConverter.java.

Öffentliche Methoden

- Document getDoc ()
- String getDocString ()
- String getURL ()

8.2.3.2 Dokumentation der Elementfunktionen

Document getDoc ()

Man muss sich nur das Dokument holen

Rückgabe:

XML Dokument für das GraphLayout Projekt

Implementiert in XmlConverter.

String getDocString ()

Implementiert in XmlConverter.

String getURL ()

Rückgabe:

Implementiert in XmlConverter.

8.2.4 Test Klassenreferenz

8.2.4.1 Ausführliche Beschreibung

Definiert in Zeile 18 der Datei Test.java.

Öffentliche, statische Methoden

- static void main (String[] args) throws InstanceException

8.2.4.2 Dokumentation der Elementfunktionen

static void main (String[] args) throws InstanceException [static]

Testen ob man Metaobjekte und Instanzen erzeugen kann und sie schließlich an die Analyse übergeben kann

Parameter:

args

Ausnahmebehandlung:

InstanceException

Definiert in Zeile 25 der Datei Test.java.

Benutzt IDataCollector.findInTree(), AnalysisFactory.getCollector(), IDataCollector.getCompleteTree(), IDataCollector.getDepth() und IDataCollector.getTree().

8.2.5 TreeObject Klassenreferenz

8.2.5.1 Ausführliche Beschreibung

Eine Repraesentation eines Ausschnittes des Metammodells und Instanzen in einer Baumstruktur

Autor:

Mart Koehler

Definiert in Zeile 14 der Datei TreeObject.java.

Öffentliche Methoden

- void addChild (DataObject child)
- ArrayList< TreeObject > getChildren ()
- DataObject getDataObject ()

- `int getLevel ()`
- `TreeObject getParent ()`
- `boolean isRoot ()`
- `void setChildren (ArrayList< TreeObject > children)`
- `void setDataObject (DataObject dataObject)`
- `TreeObject (TreeObject parent, int level)`

Private Attribute

- `ArrayList< TreeObject > children`
- `DataObject dataObject`
- `int level`
- `TreeObject parent`

8.2.5.2 Beschreibung der Konstruktoren und Destruktoren

TreeObject (TreeObject parent, int level)

Definiert in Zeile 22 der Datei `TreeObject.java`.

Wird benutzt von `TreeObject.addChild()`.

8.2.5.3 Dokumentation der Elementfunktionen

void addChild (DataObject child)

Definiert in Zeile 53 der Datei `TreeObject.java`.

Benutzt `TreeObject.getChildren()`, `TreeObject.getLevel()`, `TreeObject.setChildren()`, `TreeObject.setDataObject()` und `TreeObject.TreeObject()`.

Wird benutzt von `DataCollector.constructTree()`.

ArrayList<TreeObject> getChildren ()

Definiert in Zeile 36 der Datei `TreeObject.java`.

Benutzt `TreeObject.children`.

Wird benutzt von `TreeObject.addChild()`, `DataCollector.constructTree()`, `DataCollector.findInTree()` und `DataCollector.maxDepth()`.

DataObject getDataObject ()

Definiert in Zeile 28 der Datei `TreeObject.java`.

Benutzt `TreeObject.dataObject`.

Wird benutzt von `DataCollector.constructTree()` und `DataCollector.findInTree()`.

int getLevel ()

Definiert in Zeile 67 der Datei `TreeObject.java`.

Benutzt `TreeObject.level`.

Wird benutzt von `TreeObject.addChild()` und `DataCollector.maxDepth()`.

TreeObject getParent ()

Definiert in Zeile 72 der Datei `TreeObject.java`.

Benutzt `TreeObject.parent`.

Wird benutzt von `DataCollector.constructTree()`.

boolean isRoot ()

Definiert in Zeile 47 der Datei `TreeObject.java`.

Benutzt `TreeObject.parent`.

void setChildren (ArrayList< TreeObject > children)

Definiert in Zeile 81 der Datei `TreeObject.java`.

Wird benutzt von `TreeObject.addChild()`.

void setDataObject (DataObject dataObject)

Definiert in Zeile 32 der Datei `TreeObject.java`.

Wird benutzt von `TreeObject.addChild()`.

8.2.5.4 Dokumentation der Datenelemente

ArrayList<TreeObject> children [private]

Definiert in Zeile 19 der Datei `TreeObject.java`.

Wird benutzt von `TreeObject.getChildren()`.

DataObject dataObject [private]

Definiert in Zeile 17 der Datei `TreeObject.java`.

Wird benutzt von `TreeObject.getDataObject()`.

int level [private]

Definiert in Zeile 15 der Datei TreeObject.java.

Wird benutzt von TreeObject.getLevel().

TreeObject parent [private]

Definiert in Zeile 21 der Datei TreeObject.java.

Wird benutzt von TreeObject.getParent() und TreeObject.isRoot().

8.3 Sonstiges

8.3.1 CharacteristicAnalysis Klassenreferenz

8.3.1.1 Ausführliche Beschreibung

Klasse, die sich mit dem sammeln und aufbereiten aller von der Analyse übergebenen Daten, um daraus Kennzahlen und andere Daten zu bilden.

Autor:

Mart Koehler

Definiert in Zeile 20 der Datei CharacteristicAnalysis.java.

Öffentliche Methoden

- void addCost (DataObject obj)
- void addResponsiblePerson (String object, ArrayList< String > attribute, String[] label)
- void addToContainer (String user)
- CharacteristicAnalysis ()
- double getCost ()
- HashMap< String, ArrayList< String > > getResponsiblePerson ()
- int getServerCount ()
- int getServerDownCount ()
- String getType ()
- void setCost (double cost)
- void setServerCount (int serverCount)
- void setServerDownCount (int serverDownCount)
- void setType (String type)

Private Attribute

- double cost = 0
- HashMap< String, ArrayList< String > > responsiblePerson
- int serverCount = 0
- int serverDownCount = 0
- String type

8.3.1.2 Beschreibung der Konstruktoren und Destruktoren

CharacteristicAnalysis ()

Definiert in Zeile 27 der Datei CharacteristicAnalysis.java.

Benutzt CharacteristicAnalysis.responsiblePerson.

8.3.1.3 Dokumentation der Elementfunktionen

void addCost (DataObject obj)

Definiert in Zeile 65 der Datei CharacteristicAnalysis.java.

Benutzt CharacteristicAnalysis.getCost() und CharacteristicAnalysis.setCost().

Wird benutzt von DataCollector.constructTree().

void addResponsiblePerson (String object, ArrayList< String > attribute, String[] label)

Definiert in Zeile 100 der Datei CharacteristicAnalysis.java.

Benutzt CharacteristicAnalysis.responsiblePerson.

Wird benutzt von DataCollector.constructTree().

void addToContainer (String user)

Definiert in Zeile 76 der Datei CharacteristicAnalysis.java.

Benutzt CharacteristicAnalysis.getCost(), CharacteristicAnalysis.getResponsiblePerson(), CharacteristicAnalysis.getServerDownCount() und CharacteristicAnalysis.getType().

Wird benutzt von DataCollector.DataCollector().

double getCost ()

Definiert in Zeile 47 der Datei CharacteristicAnalysis.java.

Benutzt CharacteristicAnalysis.cost.

Wird benutzt von `CharacteristicAnalysis.addCost()` und `CharacteristicAnalysis.addToContainer()`.

HashMap<String, ArrayList<String> > getResponsiblePerson ()

Definiert in Zeile 118 der Datei `CharacteristicAnalysis.java`.

Benutzt `CharacteristicAnalysis.responsiblePerson`.

Wird benutzt von `CharacteristicAnalysis.addToContainer()`.

int getServerCount ()

Definiert in Zeile 32 der Datei `CharacteristicAnalysis.java`.

Benutzt `CharacteristicAnalysis.serverCount`.

int getServerDownCount ()

Definiert in Zeile 40 der Datei `CharacteristicAnalysis.java`.

Benutzt `CharacteristicAnalysis.serverDownCount`.

Wird benutzt von `CharacteristicAnalysis.addToContainer()`.

String getType ()

Definiert in Zeile 51 der Datei `CharacteristicAnalysis.java`.

Benutzt `CharacteristicAnalysis.type`.

Wird benutzt von `CharacteristicAnalysis.addToContainer()`.

void setCost (double cost)

Definiert in Zeile 61 der Datei `CharacteristicAnalysis.java`.

Wird benutzt von `CharacteristicAnalysis.addCost()`.

void setServerCount (int serverCount)

Definiert in Zeile 36 der Datei `CharacteristicAnalysis.java`.

void setServerDownCount (int serverDownCount)

Definiert in Zeile 44 der Datei `CharacteristicAnalysis.java`.

Wird benutzt von `DataCollector.constructTree()`.

void setType (String type)

Definiert in Zeile 56 der Datei CharacteristicAnalysis.java.

Wird benutzt von DataCollector.DataCollector().

8.3.1.4 Dokumentation der Datenelemente

double cost = 0 [private]

Definiert in Zeile 23 der Datei CharacteristicAnalysis.java.

Wird benutzt von CharacteristicAnalysis.getCost().

HashMap<String, ArrayList<String> > responsiblePerson [private]

Definiert in Zeile 24 der Datei CharacteristicAnalysis.java.

Wird benutzt von CharacteristicAnalysis.addResponsiblePerson(),
CharacteristicAnalysis.CharacteristicAnalysis() und
CharacteristicAnalysis.getResponsiblePerson().

int serverCount = 0 [private]

Definiert in Zeile 21 der Datei CharacteristicAnalysis.java.

Wird benutzt von CharacteristicAnalysis.getServerCount().

int serverDownCount = 0 [private]

Definiert in Zeile 22 der Datei CharacteristicAnalysis.java.

Wird benutzt von CharacteristicAnalysis.getServerDownCount().

String type [private]

Definiert in Zeile 25 der Datei CharacteristicAnalysis.java.

Wird benutzt von CharacteristicAnalysis.getType().

8.3.2 DataCollector Klassenreferenz

Abgeleitet von de::offis::pg::eam::mod_analysis::api::IDataCollector.

8.3.2.1 Ausführliche Beschreibung

Hier wird ein konkretete Instanz von TreeObject mit Daten aus dem Metamodell und seinen Instanzen erzeugt

Autor:

Mart Koehler

Definiert in Zeile 27 der Datei DataCollector.java.

Öffentliche Methoden

- DataCollector (HashMap< String, String > parameter, int option, ArrayList< DataObject > obj, String user, String passwd)
- boolean findInTree (DataObject obj, TreeObject tree)
- int getActualTree ()
- CharacteristicAnalysis getCharAnalysis ()
- ArrayList< TreeObject > getCompleteTree ()
- String getCostTypeObject ()
- int getDepth (TreeObject tree)
- ArrayList< DataObject > getElementsFromLevel (int level)
- ArrayList< DataObject > getObjectsFromLevel (int level)
- ArrayList< DataObject > getOccurrence ()
- int getOption ()
- String getPerson ()
- String getRelationPerson ()
- ArrayList< DataObject > getRelationsFromLevel (int level)
- ArrayList< TreeObject > getRelationTreesFromLevel (ArrayList< TreeObject > treeElem, int level)
- ArrayList< DataObject > getSearchTreeObjects ()
- TreeObject getTree (int index)
- int getTreeDepth ()
- XmlConverter getXMLConverter ()
- int maxDepth (TreeObject tree)
- void setActualTree (int actualTree)
- void setCharAnalysis (CharacteristicAnalysis charAnalysis)
- void setCostTypeObject (String costTypeObject)
- void setOccurrence (ArrayList< DataObject > occurrence)
- void setOption (int option)
- void setPerson (String person)
- void setRelationPerson (String relationPerson)
- void setSearchTree (ArrayList< TreeObject > searchTree)
- void setSearchTreeObjects (ArrayList< DataObject > searchTreeObjects)
- void setTreeDepth (int treeDepth)

Private Methoden

- void constructTree (TreeObject treeObject, String user, String passwd)
- ArrayList< TreeObject > getSearchTree ()
- void search (ArrayList< TreeObject > tree, int level)

Private Attribute

- int actualTree
- CharacteristicAnalysis charAnalysis
- String costTypeObject = "server:kosten"
- ArrayList< DataObject > occurrence
- int option
- String person = "verantwortlicher:vorname,nachname,telefon"
- String relationPerson = "hat"
- ArrayList< TreeObject > searchTree
- ArrayList< DataObject > searchTreeObjects
- ArrayList< TreeObject > tree
- int treeDepth
- XmlConverter xmlConverter

Statische private Attribute

- static Logger logger = Logger.getLogger()

8.3.2.2 Beschreibung der Konstruktoren und Destruktoren

DataCollector (HashMap< String, String > parameter, int option, ArrayList< DataObject > obj, String user, String passwd)

Konstruktor

Parameter:

- parameter** Parameter für das Setzen des Relationsnamens und Objektnamens, welche in der Kostenanalyse berücksichtigt werden
- option** 1: doppelte Werte erscheinen nicht im Abhängigkeitsbaum; 2: das erste doppelte Objekt wird in den Abhängigkeitsbaum aufgenommen und in occurrence eingetragen
- obj** Liste von Objekten, von denen aus der Baum aufgebaut werden soll
- user** der aktuelle User
- passwd** das dazugehörige Passwort

Definiert in Zeile 54 der Datei DataCollector.java.

Benutzt CharacteristicAnalysis.addToContainer(), DataCollector.charAnalysis, DataCollector.constructTree(), DataCollector.costTypeObject, DataCollector.findInTree(), DataCollector.getActualTree(), DataCollector.getCharAnalysis(), DataCollector.getCompleteTree(), DataCollector.getCostTypeObject(), DataCollector.logger, DataCollector.occurrence, DataCollector.person, DataCollector.relationPerson, DataCollector.setActualTree(), CharacteristicAnalysis.setType() und DataCollector.tree.

8.3.2.3 Dokumentation der Elementfunktionen

void constructTree (TreeObject treeObject, String user, String passwd) [private]

Definiert in Zeile 104 der Datei DataCollector.java.

Benutzt TreeObject.addChild(), CharacteristicAnalysis.addCost(), CharacteristicAnalysis.addResponsiblePerson(), DataCollector.findInTree(), DataCollector.getCharAnalysis(), TreeObject.getChildren(), DataCollector.getCompleteTree(), TreeObject.getDataObject(), DataCollector.getOccurrence(), DataCollector.getOption(), TreeObject.getParent(), DataCollector.getPerson(), DataCollector.getRelationPerson(), DataCollector.logger, CharacteristicAnalysis.setServerDownCount() und DataCollector.tree.

Wird benutzt von DataCollector.DataCollector().

boolean findInTree (DataObject obj, TreeObject tree)

Findet ein bestimmtes Object in einem Baum

Parameter:

obj Das gesuchte Objekt

tree In diesem Baum soll gesucht werden

Rückgabe:

Objekt gefunden oder nicht

Implementiert IDataCollector.

Definiert in Zeile 312 der Datei DataCollector.java.

Benutzt TreeObject.getChildren(), TreeObject.getDataObject() und DataCollector.logger.

Wird benutzt von DataCollector.constructTree() und DataCollector.DataCollector().

int getActualTree ()

Definiert in Zeile 355 der Datei DataCollector.java.

Benutzt DataCollector.actualTree.

Wird benutzt von DataCollector.DataCollector().

CharacteristicAnalysis getCharAnalysis ()

Definiert in Zeile 469 der Datei DataCollector.java.

Benutzt DataCollector.charAnalysis.

Wird benutzt von DataCollector.constructTree() und DataCollector.DataCollector().

ArrayList<TreeObject> getCompleteTree ()

Gibt alle Bäume zu allen zuvor übergebenen Objekten komplett als ArrayList zurück

Rückgabe:

ArrayList<TreeObject>

Implementiert IDataCollector.

Definiert in Zeile 338 der Datei DataCollector.java.

Benutzt DataCollector.tree.

Wird benutzt von DataCollector.constructTree(), DataCollector.DataCollector(), DataCollector.getElementsFromLevel(), DataCollector.getTree() und DataCollector.getXMLConverter().

String getCostTypeObject ()

Definiert in Zeile 516 der Datei DataCollector.java.

Benutzt DataCollector.costTypeObject.

Wird benutzt von DataCollector.DataCollector().

int getDepth (TreeObject tree)

Hauptmethode, um die maximale Tiefe eines Baumes zu erhalten

Implementiert IDataCollector.

Definiert in Zeile 463 der Datei DataCollector.java.

Benutzt DataCollector.maxDepth() und DataCollector.setTreeDepth().

ArrayList<DataObject> getElementsFromLevel (int level)

Implementiert IDataCollector.

Definiert in Zeile 392 der Datei DataCollector.java.

Benutzt `DataCollector.getCompleteTree()`, `DataCollector.getSearchTreeObjects()`, `DataCollector.search()` und `DataCollector.setSearchTreeObjects()`.

Wird benutzt von `DataCollector.getObjectsFromLevel()` und `DataCollector.getRelationsFromLevel()`.

`ArrayList<DataObject> getObjectsFromLevel (int level)`

Implementiert `IDataCollector`.

Definiert in Zeile 400 der Datei `DataCollector.java`.

Benutzt `DataCollector.getElementsFromLevel()`.

`ArrayList<DataObject> getOccurrence ()`

Implementiert `IDataCollector`.

Definiert in Zeile 498 der Datei `DataCollector.java`.

Benutzt `DataCollector.occurrence`.

Wird benutzt von `DataCollector.constructTree()`.

`int getOption ()`

Definiert in Zeile 296 der Datei `DataCollector.java`.

Benutzt `DataCollector.option`.

Wird benutzt von `DataCollector.constructTree()`.

`String getPerson ()`

Implementiert `IDataCollector`.

Definiert in Zeile 276 der Datei `DataCollector.java`.

Benutzt `DataCollector.person`.

Wird benutzt von `DataCollector.constructTree()`.

`String getRelationPerson ()`

Implementiert `IDataCollector`.

Definiert in Zeile 286 der Datei `DataCollector.java`.

Benutzt `DataCollector.relationPerson`.

Wird benutzt von `DataCollector.constructTree()`.

`ArrayList<DataObject> getRelationsFromLevel (int level)`

Implementiert IDataCollector.

Definiert in Zeile 413 der Datei DataCollector.java.

Benutzt DataCollector.getElementsFromLevel().

ArrayList<TreeObject> getRelationTreesFromLevel (ArrayList< TreeObject > treeElem, int level)

Implementiert IDataCollector.

Definiert in Zeile 426 der Datei DataCollector.java.

Benutzt DataCollector.getSearchTree(), DataCollector.search(), DataCollector.setSearchTree() und DataCollector.setSearchTreeObjects().

ArrayList<TreeObject> getSearchTree () [private]

Definiert in Zeile 383 der Datei DataCollector.java.

Benutzt DataCollector.searchTree.

Wird benutzt von DataCollector.getRelationTreesFromLevel() und DataCollector.search().

ArrayList<DataObject> getSearchTreeObjects ()

Definiert in Zeile 435 der Datei DataCollector.java.

Benutzt DataCollector.searchTreeObjects.

Wird benutzt von DataCollector.getElementsFromLevel() und DataCollector.search().

TreeObject getTree (int index)

Holt mir einen bestimmten Baum aus mehreren Bäumen

Parameter:

index Der Index des Array Elements. Die Gesamtlänge kann mit getCompleteTree ermittelt werden

Rückgabe:

ein TreeObject aus der Liste

Implementiert IDataCollector.

Definiert in Zeile 347 der Datei DataCollector.java.

Benutzt DataCollector.getCompleteTree().

int getTreeDepth ()

Definiert in Zeile 507 der Datei DataCollector.java.

Benutzt `DataCollector.treeDepth`.

Wird benutzt von `DataCollector.maxDepth()`.

XmlConverter `getXMLConverter ()`

Implementiert `IDataCollector`.

Definiert in Zeile 480 der Datei `DataCollector.java`.

Benutzt `DataCollector.getCompleteTree()` und `DataCollector.xmlConverter`.

`int maxDepth (TreeObject tree)`

Definiert in Zeile 448 der Datei `DataCollector.java`.

Benutzt `TreeObject.getChildren()`, `TreeObject.getLevel()`, `DataCollector.getTreeDepth()` und `DataCollector.setTreeDepth()`.

Wird benutzt von `DataCollector.getDepth()`.

`void search (ArrayList< TreeObject > tree, int level) [private]`

Tiefensuche nach Elementen eines bestimmten Levels auf allen Bäumen

Parameter:

tree `ArrayList` der Bäume

level Level der zu suchenden Objekte

Definiert in Zeile 368 der Datei `DataCollector.java`.

Benutzt `DataCollector.getSearchTree()` und `DataCollector.getSearchTreeObjects()`.

Wird benutzt von `DataCollector.getElementsFromLevel()` und `DataCollector.getRelationTreesFromLevel()`.

`void setActualTree (int actualTree)`

Definiert in Zeile 359 der Datei `DataCollector.java`.

Wird benutzt von `DataCollector.DataCollector()`.

`void setCharAnalysis (CharacteristicAnalysis charAnalysis)`

Definiert in Zeile 474 der Datei `DataCollector.java`.

`void setCostTypeObject (String costTypeObject)`

Definiert in Zeile 521 der Datei `DataCollector.java`.

void setOccurrence (ArrayList< DataObject > occurrence)

Definiert in Zeile 503 der Datei DataCollector.java.

void setOption (int option)

Definiert in Zeile 301 der Datei DataCollector.java.

void setPerson (String person)

Implementiert IDataCollector.

Definiert in Zeile 281 der Datei DataCollector.java.

void setRelationPerson (String relationPerson)

Implementiert IDataCollector.

Definiert in Zeile 291 der Datei DataCollector.java.

void setSearchTree (ArrayList< TreeObject > searchTree)

Definiert in Zeile 493 der Datei DataCollector.java.

Wird benutzt von DataCollector.getRelationTreesFromLevel().

void setSearchTreeObjects (ArrayList< DataObject > searchTreeObjects)

Definiert in Zeile 440 der Datei DataCollector.java.

Wird benutzt von DataCollector.getElementsFromLevel() und DataCollector.getRelationTreesFromLevel().

void setTreeDepth (int treeDepth)

Definiert in Zeile 512 der Datei DataCollector.java.

Wird benutzt von DataCollector.getDepth() und DataCollector.maxDepth().

8.3.2.4 Dokumentation der Datenelemente

int actualTree [private]

Definiert in Zeile 40 der Datei DataCollector.java.

Wird benutzt von DataCollector.getActualTree().

CharacteristicAnalysis charAnalysis [private]

Definiert in Zeile 43 der Datei DataCollector.java.

Wird benutzt von `DataCollector.DataCollector()` und `DataCollector.getCharAnalysis()`.

String costTypeObject = "server:kosten" [private]

Definiert in Zeile 30 der Datei `DataCollector.java`.

Wird benutzt von `DataCollector.DataCollector()` und `DataCollector.getCostTypeObject()`.

Logger logger = Logger.getLogger() [static, private]

Definiert in Zeile 33 der Datei `DataCollector.java`.

Wird benutzt von `DataCollector.constructTree()`, `DataCollector.DataCollector()` und `DataCollector.findInTree()`.

ArrayList<DataObject> occurrence [private]

Definiert in Zeile 34 der Datei `DataCollector.java`.

Wird benutzt von `DataCollector.DataCollector()` und `DataCollector.getOccurrence()`.

int option [private]

Definiert in Zeile 31 der Datei `DataCollector.java`.

Wird benutzt von `DataCollector.getOption()`.

String person = "verantwortlicher:vorname,nachname,telefon" [private]

Definiert in Zeile 28 der Datei `DataCollector.java`.

Wird benutzt von `DataCollector.DataCollector()` und `DataCollector.getPerson()`.

String relationPerson = "hat" [private]

Definiert in Zeile 29 der Datei `DataCollector.java`.

Wird benutzt von `DataCollector.DataCollector()` und `DataCollector.getRelationPerson()`.

ArrayList<TreeObject> searchTree [private]

Definiert in Zeile 38 der Datei `DataCollector.java`.

Wird benutzt von `DataCollector.getSearchTree()`.

ArrayList<DataObject> searchTreeObjects [private]

Definiert in Zeile 37 der Datei `DataCollector.java`.

Wird benutzt von `DataCollector.getSearchTreeObjects()`.

ArrayList<TreeObject> tree [private]

Definiert in Zeile 36 der Datei DataCollector.java.

Wird benutzt von DataCollector.constructTree(), DataCollector.DataCollector() und DataCollector.getCompleteTree().

int treeDepth [private]

Definiert in Zeile 42 der Datei DataCollector.java.

Wird benutzt von DataCollector.getTreeDepth().

XmlConverter xmlConverter [private]

Definiert in Zeile 35 der Datei DataCollector.java.

Wird benutzt von DataCollector.getXMLConverter().

8.3.3 XmlConverter Klassenreferenz

Abgeleitet von de::offis::pg::eam::mod_analysis::api::IXmlConverter.

8.3.3.1 Ausführliche Beschreibung

Die Klasse wandelt den Baum aus der Analyse in ein XML Dokument, welches mit einer entsprechenden Schnittstelle visualisiert werden kann

Autor:

Mart Koehler

Definiert in Zeile 25 der Datei XmlConverter.java.

Öffentliche Methoden

- int getDepth ()
- Document getDoc ()
- String getDocString ()
- String getURL ()
- void setDepth (int depth)
- void setDoc (Document docu)
- void setLinks (HashMap< String, String > links)
- XmlConverter (IDataCollector dataCollector, ArrayList< TreeObject > treeObject)

Private Methoden

- String `symbolName (DataObject obj)`

Private Attribute

- int `depth`
- Document `doc`
- String `docString`
- String `linkEditInstanceObject = "../extendDataInput/show_ObjectInstance.jsf?"`
- String `linkEditInstanceRelation = "../extendDataInput/show_RelationInstance.jsf?"`
- String `linkMetaObjectId = "&metaobjectId="`
- String `linkMetaRelationId = "&metarelationId="`
- String `linkObjectId = "&instanceOfMetaobject="`
- String `linkRelationId = "&instanceOfMetaobject="`

8.3.3.2 Beschreibung der Konstruktoren und Destruktoren

XmlConverter (IDataCollector dataCollector, ArrayList< TreeObject > treeObject)

Definiert in Zeile 69 der Datei `XmlConverter.java`.

Benutzt `XmlConverter.doc`, `IDataCollector.getDepth()`, `IDataCollector.getObjectsFromLevel()`, `IDataCollector.getRelationTreesFromLevel()`, `XmlConverter.linkEditInstanceObject`, `XmlConverter.linkEditInstanceRelation`, `XmlConverter.linkMetaObjectId`, `XmlConverter.linkMetaRelationId`, `XmlConverter.linkObjectId`, `XmlConverter.linkRelationId`, `XmlConverter.setDoc()` und `XmlConverter.symbolName()`.

8.3.3.3 Dokumentation der Elementfunktionen

int getDepth ()

Definiert in Zeile 259 der Datei `XmlConverter.java`.

Benutzt `XmlConverter.depth`.

Document getDoc ()

Implementiert `IXmlConverter`.

Definiert in Zeile 284 der Datei `XmlConverter.java`.

Benutzt `XmlConverter.doc`.

String getDocString ()

Implementiert IXmlConverter.

Definiert in Zeile 252 der Datei XmlConverter.java.

Benutzt XmlConverter.docString.

String getURL ()

Implementiert IXmlConverter.

Definiert in Zeile 277 der Datei XmlConverter.java.

void setDepth (int depth)

Definiert in Zeile 263 der Datei XmlConverter.java.

void setDoc (Document docu)

Definiert in Zeile 269 der Datei XmlConverter.java.

Benutzt XmlConverter.doc.

Wird benutzt von XmlConverter.XmlConverter().

void setLinks (HashMap< String, String > links)

Wenn links in der XML von der Vorgabe abweichen bzw von einer anderen Quelle kommen, dann können sie hier verändert werden

Parameter:

links

Definiert in Zeile 46 der Datei XmlConverter.java.

Benutzt XmlConverter.linkEditInstanceObject, XmlConverter.linkEditInstanceRelation, XmlConverter.linkMetaObjectId, XmlConverter.linkMetaRelationId, XmlConverter.linkObjectId und XmlConverter.linkRelationId.

String symbolName (DataObject obj) [private]

Definiert in Zeile 230 der Datei XmlConverter.java.

Wird benutzt von XmlConverter.XmlConverter().

8.3.3.4 Dokumentation der Datenelemente

int depth [private]

Definiert in Zeile 27 der Datei XmlConverter.java.

Wird benutzt von `XmlConverter.getDepth()`.

Document doc [private]

Definiert in Zeile 28 der Datei `XmlConverter.java`.

Wird benutzt von `XmlConverter.getDoc()`, `XmlConverter.setDoc()` und `XmlConverter.XmlConverter()`.

String docString [private]

Definiert in Zeile 29 der Datei `XmlConverter.java`.

Wird benutzt von `XmlConverter.getDocString()`.

String linkEditInstanceObject = "../extendDataInput/show_ObjectInstance.jsf?"
[private]

Definiert in Zeile 35 der Datei `XmlConverter.java`.

Wird benutzt von `XmlConverter.setLinks()` und `XmlConverter.XmlConverter()`.

String linkEditInstanceRelation = "../extendDataInput/show_RelationInstance.jsf?"
[private]

Definiert in Zeile 39 der Datei `XmlConverter.java`.

Wird benutzt von `XmlConverter.setLinks()` und `XmlConverter.XmlConverter()`.

String linkMetaObjectId = "&metaobjectId=" [private]

Definiert in Zeile 33 der Datei `XmlConverter.java`.

Wird benutzt von `XmlConverter.setLinks()` und `XmlConverter.XmlConverter()`.

String linkMetaRelationId = "&metarelationId=" [private]

Definiert in Zeile 37 der Datei `XmlConverter.java`.

Wird benutzt von `XmlConverter.setLinks()` und `XmlConverter.XmlConverter()`.

String linkObjectId = "&instanceOfMetaobject=" [private]

Definiert in Zeile 34 der Datei `XmlConverter.java`.

Wird benutzt von `XmlConverter.setLinks()` und `XmlConverter.XmlConverter()`.

String linkRelationId = "&instanceOfMetaobject=" [private]

Definiert in Zeile 38 der Datei `XmlConverter.java`.

Wird benutzt von `XmlConverter.setLinks()` und `XmlConverter.XmlConverter()`.

9 de.offis.pg.eam.mod_vis_1

9.1 BundleConfiguration

9.1.1 BundleViews Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::module::api::IModuleView.

9.1.1.1 Ausführliche Beschreibung

Definiert in Zeile 22 der Datei BundleViews.java.

Öffentliche Methoden

- String getBundleVersion ()
- String getDescription ()
- long getEquinoxBundleId ()
- String getName ()
- List< View > getViews ()

Statische private Attribute

- static final String VIEWS = "bundle_views.xml"

9.1.1.2 Dokumentation der Elementfunktionen

String getBundleVersion ()

Definiert in Zeile 68 der Datei BundleViews.java.

String getDescription ()

Definiert in Zeile 28 der Datei BundleViews.java.

long getEquinoxBundleId ()

Definiert in Zeile 33 der Datei BundleViews.java.

String getName ()

Definiert in Zeile 38 der Datei BundleViews.java.

List<View> getViews ()

Definiert in Zeile 43 der Datei BundleViews.java.

Benutzt BundleViews.VIEWS.

9.1.1.3 Dokumentation der Datenelemente

final String VIEWS = "bundle_views.xml" [static, private]

Definiert in Zeile 24 der Datei BundleViews.java.

Wird benutzt von BundleViews.getViews().

9.1.2 Configuration Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::configuration::api::IBundleConfiguration.

9.1.2.1 Ausführliche Beschreibung

Definiert in Zeile 10 der Datei Configuration.java.

Öffentliche Methoden

- boolean bundleHasObjects ()
- boolean bundleHasRelations ()
- boolean bundleHasViews ()
- Document getObjectsJdomDocument () throws JDOMException, IOException

9.1.2.2 Dokumentation der Elementfunktionen

boolean bundleHasObjects ()

Definiert in Zeile 13 der Datei Configuration.java.

boolean bundleHasRelations ()

Definiert in Zeile 28 der Datei Configuration.java.

boolean bundleHasViews ()

Definiert in Zeile 18 der Datei Configuration.java.

Document getObjectsJdomDocument () throws JDOMException, IOException

Definiert in Zeile 23 der Datei Configuration.java.

9.1.3 Activator Klassenreferenz

Abgeleitet von org::osgi::framework::BundleActivator.

9.1.3.1 Ausführliche Beschreibung

Definiert in Zeile 30 der Datei Activator.java.

Öffentliche Methoden

- void start (BundleContext context) throws Exception
- void stop (BundleContext context) throws Exception

Statische öffentliche Attribute

- static String bundleSymbolicName
- static String bundleVersion
- static long equinoxBundleId

Private Attribute

- ServiceTracker httpServiceTracker

Klassen

- class HttpServiceTracker
- class ServletContextListenerServletAdaptor

9.1.3.2 Dokumentation der Elementfunktionen**void start (BundleContext context) throws Exception**

Definiert in Zeile 38 der Datei Activator.java.

Benutzt Activator.bundleSymbolicName, Activator.bundleVersion, Activator.equinoxBundleId und Activator.httpServiceTracker.

void stop (BundleContext context) throws Exception

Definiert in Zeile 47 der Datei Activator.java.

Benutzt Activator.httpServiceTracker.

9.1.3.3 Dokumentation der Datenelemente

String bundleSymbolicName [static]

Definiert in Zeile 34 der Datei Activator.java.

Wird benutzt von Activator.start().

String bundleVersion [static]

Definiert in Zeile 35 der Datei Activator.java.

Wird benutzt von Activator.start().

long equinoxBundleId [static]

Definiert in Zeile 36 der Datei Activator.java.

Wird benutzt von Activator.start().

ServiceTracker httpServiceTracker [private]

Definiert in Zeile 32 der Datei Activator.java.

Wird benutzt von Activator.start() und Activator.stop().

9.1.4 Activator.HttpServiceTracker Klassenreferenz

Abgeleitet von org::osgi::util::tracker::ServiceTracker.

9.1.4.1 Ausführliche Beschreibung

Definiert in Zeile 51 der Datei Activator.java.

Öffentliche Methoden

- Object addingService (ServiceReference reference)
- HttpServiceTracker (BundleContext context)
- void removedService (ServiceReference reference, Object service)

Statische private Attribute

- static final String PATH = "/" + Constants.MODULE_NAME
- static final String WEB_ROOT = "/WebRoot"

9.1.4.2 Beschreibung der Konstruktoren und Destruktoren

HttpServiceTracker (BundleContext context)

Definiert in Zeile 56 der Datei Activator.java.

9.1.4.3 Dokumentation der Elementfunktionen

Object addingService (ServiceReference reference)

Definiert in Zeile 60 der Datei Activator.java.

Benutzt Activator.HttpServiceTracker.PATH und Activator.HttpServiceTracker.WEB_ROOT.

void removedService (ServiceReference reference, Object service)

Definiert in Zeile 96 der Datei Activator.java.

Benutzt Activator.HttpServiceTracker.PATH.

9.1.4.4 Dokumentation der Datenelemente

final String PATH = "/" + Constants.MODULE_NAME [static, private]

Definiert in Zeile 53 der Datei Activator.java.

Wird benutzt von Activator.HttpServiceTracker.addingService() und Activator.HttpServiceTracker.removedService().

final String WEB_ROOT = "/WebRoot" [static, private]

Definiert in Zeile 54 der Datei Activator.java.

Wird benutzt von Activator.HttpServiceTracker.addingService().

9.1.5 Activator.ServletContextListenerServletAdaptor Klassenreferenz

Abgeleitet von javax::servlet::Servlet.

9.1.5.1 Ausführliche Beschreibung

Definiert in Zeile 105 der Datei Activator.java.

Öffentliche Methoden

- void destroy ()
- ServletConfig getServletConfig ()
- String getServletInfo ()
- void init (ServletConfig config) throws ServletException
- void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException
- ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Private Attribute

- ServletConfig config
- Servlet delegate
- ClassLoader jspLoader
- ServletContextListener listener

9.1.5.2 Beschreibung der Konstruktoren und Destruktoren

ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Definiert in Zeile 111 der Datei Activator.java.

9.1.5.3 Dokumentation der Elementfunktionen

void destroy ()

Definiert in Zeile 145 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config, Activator.ServletContextListenerServletAdaptor.delegate, Activator.ServletContextListenerServletAdaptor.jspLoader und Activator.ServletContextListenerServletAdaptor.listener.

ServletConfig getServletConfig ()

Definiert in Zeile 159 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config.

String getServletInfo ()

Definiert in Zeile 163 der Datei Activator.java.

void init (ServletConfig config) throws ServletException

Definiert in Zeile 119 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config,
Activator.ServletContextListenerServletAdaptor.delegate,
Activator.ServletContextListenerServletAdaptor.jspLoader und
Activator.ServletContextListenerServletAdaptor.listener.

void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException

Definiert in Zeile 133 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.delegate und
Activator.ServletContextListenerServletAdaptor.jspLoader.

9.1.5.4 Dokumentation der Datenelemente**ServletConfig config** [private]

Definiert in Zeile 106 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.getServletConfig() und
Activator.ServletContextListenerServletAdaptor.init().

Servlet delegate [private]

Definiert in Zeile 108 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.init() und
Activator.ServletContextListenerServletAdaptor.service().

ClassLoader jspLoader [private]

Definiert in Zeile 109 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.init() und
Activator.ServletContextListenerServletAdaptor.service().

ServletContextListener listener [private]

Definiert in Zeile 107 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy() und
Activator.ServletContextListenerServletAdaptor.init().

9.1.6 ModVis1MenuContributor Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::api::AbstractAdoptableMenuContributor.

9.1.6.1 Ausführliche Beschreibung

Definiert in Zeile 6 der Datei ModVis1MenuContributor.java.

Öffentliche Methoden

- String getAdopterId ()
- String getContributionTarget ()
- String getContributorId ()
- String getTargetSubmenu ()
- ModVis1MenuContributor ()

9.1.6.2 Beschreibung der Konstruktoren und Destruktoren

ModVis1MenuContributor ()

Definiert in Zeile 8 der Datei ModVis1MenuContributor.java.

9.1.6.3 Dokumentation der Elementfunktionen

String getAdopterId ()

Definiert in Zeile 12 der Datei ModVis1MenuContributor.java.

String getContributionTarget ()

Definiert in Zeile 16 der Datei ModVis1MenuContributor.java.

String getContributorId ()

Definiert in Zeile 20 der Datei ModVis1MenuContributor.java.

String getTargetSubmenu ()

Definiert in Zeile 24 der Datei ModVis1MenuContributor.java.

9.2 Schnittstelle zur Analyse

9.2.1 AnalysisAdapter Klassenreferenz

Abgeleitet von `de::offis::pg::eam::mod_vis_1::adapter::IAnalysisAdapter`.

9.2.1.1 Ausführliche Beschreibung

Adapter Klasse, um eine "Schnittstelle" zum Analysemodul zu haben. Außerdem lässt sich von hier aus einfach auf Analysemethoden zugreifen.

Autor:

Mart Koehler

Definiert in Zeile 20 der Datei `AnalysisAdapter.java`.

Öffentliche Methoden

- `AnalysisAdapter (HashMap< String, String > parameter, ArrayList< DataObject > obj)`
- `IDataCollector getCollector ()`
- `IXmlConverter getConverter ()`

Private Attribute

- `IDataCollector collector`
- `IXmlConverter converter`
- `AnalysisFactory factory`

9.2.1.2 Beschreibung der Konstruktoren und Destrukturen

`AnalysisAdapter (HashMap< String, String > parameter, ArrayList< DataObject > obj)`

Definiert in Zeile 25 der Datei `AnalysisAdapter.java`.

Benutzt `AnalysisAdapter.collector`, `AnalysisAdapter.converter` und `AnalysisAdapter.factory`.

9.2.1.3 Dokumentation der Elementfunktionen

`IDataCollector getCollector ()`

Implementiert `IAnalysisAdapter`.

Definiert in Zeile 32 der Datei AnalysisAdapter.java.

Benutzt AnalysisAdapter.collector.

Wird benutzt von TreeVisualization.drawNewSVG() und MatrixVisualization.drawNewSVG().

IXmlConverter getConverter ()

Implementiert IAnalysisAdapter.

Definiert in Zeile 38 der Datei AnalysisAdapter.java.

Benutzt AnalysisAdapter.converter.

9.2.1.4 Dokumentation der Datenelemente

IDataCollector collector [private]

Definiert in Zeile 22 der Datei AnalysisAdapter.java.

Wird benutzt von AnalysisAdapter.AnalysisAdapter() und AnalysisAdapter.getCollector().

IXmlConverter converter [private]

Definiert in Zeile 23 der Datei AnalysisAdapter.java.

Wird benutzt von AnalysisAdapter.AnalysisAdapter() und AnalysisAdapter.getConverter().

AnalysisFactory factory [private]

Definiert in Zeile 21 der Datei AnalysisAdapter.java.

Wird benutzt von AnalysisAdapter.AnalysisAdapter().

9.2.2 IAnalysisAdapter Schnittstellenreferenz

Basisklasse für AnalysisAdapter.

9.2.2.1 Ausführliche Beschreibung

Definiert in Zeile 7 der Datei IAnalysisAdapter.java.

Öffentliche Methoden

- IDataCollector getCollector ()

- `IXmlConverter getConverter ()`

9.2.2.2 Dokumentation der Elementfunktionen

IDataCollector getCollector ()

Implementiert in `AnalysisAdapter`.

IXmlConverter getConverter ()

Implementiert in `AnalysisAdapter`.

9.3 Hilfsklassen

9.3.1 Constants Klassenreferenz

9.3.1.1 Ausführliche Beschreibung

Autor:

philipp In dieser Klasse werden Konstanten definiert und gesammelt, die für das Visualisierungsmodul `ModVis1` benutzt werden.

Definiert in Zeile 11 der Datei `Constants.java`.

Statische öffentliche Attribute

- `static final String CONFIG_FILE_NAME = "config_eam_mod_vis.xml"`
- `static final String MODULE_NAME = "ModVis1"`
- `static final long OLD_FILE = 86400000`
- `static final String PATH_TO_CONFIG_FILE = System.getProperty("user.dir")+SEP+"eamconfig"+SEP+CONFIG_FILE_NAME`
- `static final String PATH_TO_IMAGES = "images"`
- `static final String PNG_FOLDER_NAME = "png"`
- `static final String SEP = System.getProperty("file.separator")`
- `static final String SVG_FOLDER_NAME = "svg"`

9.3.1.2 Dokumentation der Datenelemente

`final String CONFIG_FILE_NAME = "config_eam_mod_vis.xml" [static]`

Gibt den Namen der Datei an in der der Pfad zu `ModVis1` und `ModVis2` hinterlegt ist

Definiert in Zeile 20 der Datei `Constants.java`.

final String MODULE_NAME = "ModVis1" [static]

Modulname im Webkontext, also der Name des Ordners Webroot...

Definiert in Zeile 14 der Datei Constants.java.

final long OLD_FILE = 86400000 [static]

Dateien, die älter sind, werden gelöscht, sobald eine neue Datei in den Container gelegt wird. wird nur bei von ModVis1 generierten (svg / jpg) dateien gemacht

Definiert in Zeile 41 der Datei Constants.java.

**final String PATH_TO_CONFIG_FILE =
System.getProperty("user.dir")+SEP+"eamconfig"+SEP+CONFIG_FILE_NAME
[static]**

Gibt den Pfad (inkl. Dateiname) an unter der die Konfigurationsdatei mit den Pfaden zu ModVis1 / ModVis2 zu finden ist.

Definiert in Zeile 24 der Datei Constants.java.

final String PATH_TO_IMAGES = "images" [static]

Gibt den Pfad zu dem Ordner an in dem die generierten Bilder (SVG, JPG) gespeichert werden.

Definiert in Zeile 28 der Datei Constants.java.

final String PNG_FOLDER_NAME = "png" [static]

So heisst der Ordner in /eamconfig/ModVis1/images in dem die PNGs abgelegt werden sollen

Definiert in Zeile 36 der Datei Constants.java.

final String SEP = System.getProperty("file.separator") [static]

Erzeugt das Zeichen zum Trennen von Ordnern "/" (Linux) oder "\" (Windows)

Definiert in Zeile 17 der Datei Constants.java.

final String SVG_FOLDER_NAME = "svg" [static]

So heisst der Ordner in /eamconfig/ModVis1/images in dem die SVGs abgelegt werden sollen

Definiert in Zeile 32 der Datei Constants.java.

9.3.2 Definitions Klassenreferenz

9.3.2.1 Ausführliche Beschreibung

Diese Klasse stellt Methoden zur Verfügung, die den `<defs> ... </defs>` Bereich in einem SVG setzen. Auf die definierten Objekte kann dann per `<use> ... </use>` zugegriffen werden.

Autor:

philipp

Definiert in Zeile 20 der Datei Definitions.java.

Öffentliche, statische Methoden

- static void beginDefinition (XMLEventWriter writer)
- static void checkMarkDefinition (XMLEventWriter writer)
- static void completeDefBlock (XMLEventWriter writer)
- static void crossMarkDefinition (XMLEventWriter writer)
- static void endDefinition (XMLEventWriter writer)
- static void myComputerDefinition (XMLEventWriter writer)
- static void myGeschProzDefinition (XMLEventWriter writer)
- static void myObjectDefinition (XMLEventWriter writer)
- static void myServerDefinition (XMLEventWriter writer)
- static void mySoftwareDefinition (XMLEventWriter writer)
- static void selfMarkDefinition (XMLEventWriter writer)
- static void toolTipScript (XMLEventWriter writer)

9.3.2.2 Dokumentation der Elementfunktionen

static void beginDefinition (XMLEventWriter writer) [static]

Diese Methode erzeugt das öffnende `<defs>`-Tag in einem SVG. Zu beachten ist, dass diese Methode an der richtigen Stelle aufgerufen wird, damit der `<defs>...</defs>`-Bereich an der gewünschten Stelle im SVG auftaucht.

Parameter:

writer In diese Datei wird das Tag geschrieben.

Definiert in Zeile 58 der Datei Definitions.java.

Wird benutzt von Definitions.completeDefBlock().

static void checkMarkDefinition (XMLEventWriter writer) [static]

Hier wird festgelegt, wie das Symbol aussehen soll, dass veranschaulichen soll, dass zwei Objekte miteinander Verbunden sind (grüner Haken). Zu Beachten ist, dass die Methode nach `Definitions.beginDefinition()` und vor `Definitions.endDefinition()` aufgerufen werden muss.

Parameter:

writer In dieser Datei wird das Symbol definiert.

Definiert in Zeile 112 der Datei `Definitions.java`.

Wird benutzt von `Definitions.completeDefBlock()`.

static void completeDefBlock (XMLEventWriter writer) [static]

Diese Methode schreibt den gesamten `<defs></defs>` Block in die Datei.

Parameter:

writer In diese Datei soll der Block geschrieben werden

Siehe auch:

`Definitions.beginDefinition`
`Definitions.checkMarkDefinition`
`Definitions.crossMarkDefinition`
`Definitions.myComputerDefinition`
`Definitions.myObjectDefinition`
`Definitions.myServerDefinition`
`Definitions.mySoftwareDefinition`
`Definitions.selfMarkDefinition`
`Definitions.endDefinition`

Definiert in Zeile 36 der Datei `Definitions.java`.

Benutzt `Definitions.beginDefinition()`, `Definitions.checkMarkDefinition()`, `Definitions.crossMarkDefinition()`, `Definitions.endDefinition()`, `Definitions.myComputerDefinition()`, `Definitions.myGeschProzDefinition()`, `Definitions.myObjectDefinition()`, `Definitions.myServerDefinition()`, `Definitions.mySoftwareDefinition()`, `Definitions.selfMarkDefinition()` und `Definitions.toolTipScript()`.

static void crossMarkDefinition (XMLEventWriter writer) [static]

Hier wird festgelegt, wie das Symbol aussehen soll, dass veranschaulichen soll, dass zwei Objekte NICHT miteinander Verbunden sind (rotes Kreuz). Zu Beachten ist, dass die Methode nach `Definitions.beginDefinition()` und vor `Definitions.endDefinition()` aufgerufen werden muss.

Parameter:

writer In dieser Datei wird das Symbol definiert.

Definiert in Zeile 140 der Datei `Definitions.java`.

Wird benutzt von `Definitions.completeDefBlock()`.

static void endDefinition (XMLEventWriter writer) [static]

Diese Methode erzeugt das schliessende `</defs>`-Tag in einem SVG. Zu beachten ist, dass diese Methode an der richtigen Stelle aufgerufen wird, damit der `<defs>...</defs>`-Bereich an der gewünschten Stelle im SVG auftaucht.

Parameter:

writer In diese Datei wird das Tag geschrieben.

Definiert in Zeile 590 der Datei `Definitions.java`.

Wird benutzt von `Definitions.completeDefBlock()`.

static void myComputerDefinition (XMLEventWriter writer) [static]

Hier wird festgelegt, wie das Symbol aussehen soll, dass einen Computer (Workstation/Arbeitsplatzrechner) darstellt. Zu Beachten ist, dass die Methode nach `Definitions.beginDefinition()` und vor `Definitions.endDefinition()` aufgerufen werden muss.

Parameter:

writer In dieser Datei wird das Symbol definiert.

Definiert in Zeile 330 der Datei `Definitions.java`.

Wird benutzt von `Definitions.completeDefBlock()`.

static void myGeschProzDefinition (XMLEventWriter writer) [static]

Hier wird festgelegt, wie das Symbol aussehen soll, dass einen Geschäftsprozess darstellt.

Parameter:

writer In dieser Datei wird das Symbol definiert.

Definiert in Zeile 197 der Datei `Definitions.java`.

Wird benutzt von `Definitions.completeDefBlock()`.

static void myObjectDefinition (XMLEventWriter writer) [static]

Hier wird festgelegt, wie das Symbol aussehen soll, dass ein nicht näher bekanntes EAM-Objekt darstellt. Zu Beachten ist, dass die Methode nach `Definitions.beginDefinition()` und vor `Definitions.endDefinition()` aufgerufen werden muss.

Parameter:

writer In dieser Datei wird das Symbol definiert.

Definiert in Zeile 538 der Datei `Definitions.java`.

Wird benutzt von `Definitions.completeDefBlock()`.

static void myServerDefinition (XMLEventWriter writer) [static]

Hier wird festgelegt, wie das Symbol aussehen soll, dass einen Server darstellt. Zu Beachten ist, dass die Methode nach `Definitions.beginDefinition()` und vor `Definitions.endDefinition()` aufgerufen werden muss.

Parameter:

writer In dieser Datei wird das Symbol definiert.

Definiert in Zeile 230 der Datei `Definitions.java`.

Wird benutzt von `Definitions.completeDefBlock()`.

static void mySoftwareDefinition (XMLEventWriter writer) [static]

Hier wird festgelegt, wie das Symbol aussehen soll, dass eine Software darstellt. Zu Beachten ist, dass die Methode nach `Definitions.beginDefinition()` und vor `Definitions.endDefinition()` aufgerufen werden muss.

Parameter:

writer In dieser Datei wird das Symbol definiert.

Definiert in Zeile 406 der Datei `Definitions.java`.

Wird benutzt von `Definitions.completeDefBlock()`.

static void selfMarkDefinition (XMLEventWriter writer) [static]

Hier wird festgelegt, wie das Symbol aussehen soll, dass veranschaulichen soll, dass ein Objekt mit sich selbst in Verbindung steht - oder auch eben nicht. Jenachdem, wie man das lesen möchte. Zu Beachten ist, dass die Methode nach `Definitions.beginDefinition()` und vor `Definitions.endDefinition()` aufgerufen werden muss.

Parameter:

writer In dieser Datei wird das Symbol definiert.

Definiert in Zeile 169 der Datei `Definitions.java`.

Wird benutzt von `Definitions.completeDefBlock()`.

static void toolTipScript (XMLEventWriter writer) [static]

Diese Methode schreibt die JavaScript-Funktionen in den defs-Block. Diese Funktionen müssen in `/eamconfig/ModVis1/script.txt` liegen.

Parameter:

writer In diese Datei wird das Script geschrieben.

Definiert in Zeile 75 der Datei `Definitions.java`.

Wird benutzt von `Definitions.completeDefBlock()`.

9.3.3 ExportSVG Klassenreferenz

9.3.3.1 Ausführliche Beschreibung

Wandelt ein `svgBild` in ein anders Bildformat um. Das neue Bild findet sich im Gleichen Ordner wie das alte Bild, es sei denn, man gibt den Methoden einen neuen Speicherplatz mit.

Autor:

philipp

Definiert in Zeile 23 der Datei `ExportSVG.java`.

Öffentliche, statische Methoden

- `static void toJPG (String pathToSVGImage, String pathToJPGImage, String jpgName)`
- `static void toJPG (String pathToSVGImage, String pathToJPGImage)`
- `static void toJPG (String pathToSVGImage)`
- `static void toPNG (String file1, String file2)`

Statische private Attribute

- `static String separator = Constants.SEP`

9.3.3.2 Dokumentation der Elementfunktionen

`static void toJPG (String pathToSVGImage, String pathToJPGImage, String jpgName)` `[static]`

Erzeugt aus einem SVG-Bild ein JPG-Bild an einem benutzerdefinierten Ort

Parameter:

`pathToSVGImage` Hier liegt das SVG-Bild (Voller Pfad)

`pathToJPGImage` Hier soll das JPG-Bild liegen

`jpgName` Den Namen soll das neue JPG haben

Definiert in Zeile 99 der Datei `ExportSVG.java`.

Benutzt `ExportSVG.separator`.

`static void toJPG (String pathToSVGImage, String pathToJPGImage)` `[static]`

Erzeugt aus einem SVG-Bild ein JPG-Bild an einem benutzerdefinierten Ort

Parameter:

pathToSVGImage Hier liegt das SVG-Bild (Voller Pfad)

pathToJPGImage Hier soll das JPG-Bild liegen (Voller Pfad mit Bild-Namen)

Definiert in Zeile 66 der Datei ExportSVG.java.

static void toJPG (String pathToSVGImage) [static]

Erzeugt aus einem SVG-Bild ein JPG-Bild im gleichen Ordner mit gleichem Namen wie das SVG.

Parameter:

pathToSVGImage Hier liegt das SVG-Bild (Voller Pfad)

Definiert in Zeile 33 der Datei ExportSVG.java.

static void toPNG (String file1, String file2) [static]

Exportiert ein SVG-Bild in ein PNG-Bild. Bei den Pfaden müssen die vollständigen Pfade (inkl Dateinamen und Endung) angegeben werden.

Parameter:

file1 Pfad zum SVG-Bild

file2 Pfad zum PNG-Bild

Definiert in Zeile 137 der Datei ExportSVG.java.

9.3.3.3 Dokumentation der Datenelemente

String separator = Constants.SEP [static, private]

Definiert in Zeile 25 der Datei ExportSVG.java.

Wird benutzt von ExportSVG.toJPG().

9.3.4 StaxUtil Klassenreferenz

9.3.4.1 Ausführliche Beschreibung

In dieser Klasse sind die Methoden gesammelt, die das Zeichnen von SVGs erleichtern.

Autor:

philipp

Definiert in Zeile 26 der Datei StaxUtil.java.

Öffentliche, statische Methoden

- static void drawCheckMark (XMLEventWriter writer, float x, float y, float width, float height)
- static void drawCrossMark (XMLEventWriter writer, float x, float y, float width, float height)
- static void drawDataObject (XMLEventWriter writer, DataObject dataObject, float x, float y, float width, float height)
- static void drawEdgeWithInfo (TreeObject relation, XMLEventWriter writer, float x1, float y1, float x2, float y2, String info)
- static void drawLine (XMLEventWriter writer, float x1, float y1, float x2, float y2)
- static void drawSelfMark (XMLEventWriter writer, float x, float y, float width, float height)
- static void svgFooter (XMLEventWriter writer)
- static void svgHeaderWithScript (XMLEventWriter writer, float width, float height, float viewBoxWidth, float viewBoxHeight, String color)
- static void svgPrettyFormat (String pathAndSvgName)
- static void svgPrettyFormatTree (String pathAndSvgName)
- static void svgToolTipGroup (XMLEventWriter writer)

Statische private Attribute

- static final String linkEditInstanceObject =
"../extendDataInput/show_ObjectInstance.jsf?"
- static final String linkEditInstanceRelation =
"../extendDataInput/show_RelationInstance.jsf?"
- static final String linkMetaObjectId = "&metaobjectId="
- static final String linkMetaRelationId = "&metarelationId="
- static final String linkObjectId = "&instanceOfMetaobject="
- static final String linkRelationId = "&instanceOfMetaobject="

9.3.4.2 Dokumentation der Elementfunktionen

static void drawCheckMark (XMLEventWriter writer, float x, float y, float width, float height) [static]

Diese Methode zeichnet das Symbol, das dafür steht, dass ein Objekt mit einem anderen verbunden ist. Wie genau das Symbol aussieht (zB grüner Haken), ist in `Definitions.checkMarkDefinition()` festgelegt.

Siehe auch:

`Definitions.checkMarkDefinition()`

Parameter:

- writer** In diese Datei soll das Symbol gezeichnet werden
- x** x-Koordinate der linken oberen Ecke des Symbols.
- y** y-Koordinate der linken oberen Ecke des Symbols.
- width** Breite des Symbols.
- height** Höhe des Symbols.

Definiert in Zeile 231 der Datei StaxUtil.java.

static void drawCrossMark (XMLEventWriter writer, float x, float y, float width, float height) [static]

Diese Methode zeichnet das Symbol, das dafür steht, dass ein Objekt NICHT mit einem anderen verbunden ist. Wie genau das Symbol aussieht (zB rotes X), ist in Definitions.crossMarkDefinition() festgelegt.

Siehe auch:

Definitions.crossMarkDefinition()

Parameter:

- writer** In diese Datei soll das Symbol gezeichnet werden
- x** x-Koordinate der linken oberen Ecke des Symbols.
- y** y-Koordinate der linken oberen Ecke des Symbols.
- width** Breite des Symbols.
- height** Höhe des Symbols.

Definiert in Zeile 201 der Datei StaxUtil.java.

static void drawDataObject (XMLEventWriter writer, DataObject dataObject, float x, float y, float width, float height) [static]

Diese Methode malt "echte" Objekte (im Sinne von DataObject), keine Relationen. Obwohl Relationen auch DataObjects sind, sollte an diese Methode nur ein echtes Objekt übergeben werden. Also solche, die dataObject.isObject == true liefern.

Parameter:

- writer** In diese Datei soll das Object gemalt werden.
- dataObject** Dieses Object soll dargestellt werden.
- x** x-Koordinate der linken oberen Ecke des Objektes.
- y** y-Koordinate der linken oberen Ecke des Objektes.
- width** Breite des Objektes.
- height** Höhe des Objektes.

Definiert in Zeile 98 der Datei StaxUtil.java.

Benutzt StaxUtil.linkEditInstanceObject, StaxUtil.linkMetaObjectId und StaxUtil.linkObjectId.

static void drawEdgeWithInfo (TreeObject relation, XMLEventWriter writer, float x1, float y1, float x2, float y2, String info) [static]

Diese Methode zeichnet eine Linie in ein SVG ein. Es wird auch ein Tool- Tip an der Linie angezeigt (sofern das richtige JavaScript richtig eingebunden wurde).

Parameter:

relation Diese Relation wird durch die Linie im Baum repräsentiert

writer In diese Datei soll die Linie geschrieben werden.

x1 x-Koordinate der Startposition.

y1 y-Koordinate der Startposition.

x2 x-Koordinate der Endposition.

y2 y-Koordinate der Endposition.

info Diese Info wird als ToolTip an die Linie gegeben.

Definiert in Zeile 413 der Datei StaxUtil.java.

Benutzt StaxUtil.linkEditInstanceRelation, StaxUtil.linkMetaRelationId und StaxUtil.linkRelationId.

static void drawLine (XMLEventWriter writer, float x1, float y1, float x2, float y2) [static]

Diese Methode zeichnet eine schlichte Linie in ein SVG ein.

Parameter:

writer In diese Datei soll die Linie geschrieben werden.

x1 x-Koordinate der Startposition.

y1 y-Koordinate der Startposition.

x2 x-Koordinate der Endposition.

y2 y-Koordinate der Endposition.

Definiert in Zeile 460 der Datei StaxUtil.java.

static void drawSelfMark (XMLEventWriter writer, float x, float y, float width, float height) [static]

Diese Methode zeichnet das Symbol, das dafür steht, dass ein Objekt mit sich selbst in Verbindung steht - oder auch eben nicht. Jenachdem, wie man das lesen möchte. Wie genau das Symbol aussieht, ist in Definitions.selfMarkDefinition() festgelegt.

Siehe auch:

Definitions.selfMarkDefinition()

Parameter:

writer In diese Datei soll das Symbol gezeichnet werden

x x-Koordinate der linken oberen Ecke des Symbols.

y y-Koordinate der linken oberen Ecke des Symbols.

width Breite des Symbols.

height Höhe des Symbols.

Definiert in Zeile 169 der Datei StaxUtil.java.

static void svgFooter (XMLEventWriter writer) [static]

Diese Methode schreibt den Footer einer SVG-Datei (</svg>).

Parameter:

writer In diese Datei soll der Footer geschrieben werden.

Definiert in Zeile 312 der Datei StaxUtil.java.

static void svgHeaderWithScript (XMLEventWriter writer, float width, float height, float viewBoxWidth, float viewBoxHeight, String color) [static]

Diese Methode setzt den Header der SVG-Datei.

Parameter:

writer In diese Datei wird der Header geschrieben.

width Die Breite des SVG-Bildes.

height Die Höhe des SVG-Bildes.

viewBoxWidth Die Breite der ViewBox.

viewBoxHeight Die Höhe der ViewBox.

color Hintergrundfarbe in der Form "#e6e6e6" - kann auch "" sein, dann Defaultfarbe: hellgrau

Definiert in Zeile 48 der Datei StaxUtil.java.

static void svgPrettyFormat (String pathAndSvgName) [static]

Diese Methode bekommt eine vorhandene .svg-Datei übergeben und trägt dann Sorge, dass die Datei menschenlesbar formatiert wird (Einrücken, Umbrüche, etc.). Des Weiteren wird auch der DocType in die Datei geschrieben.

Parameter:

pathAndSvgName Der vollständige Pfad zur zu formatierenden Datei.

Definiert in Zeile 334 der Datei StaxUtil.java.

static void svgPrettyFormatTree (String pathAndSvgName) [static]

Diese Methode bekommt eine vorhandene .svg-Datei (die einen generierten Baum enthält) übergeben und trägt dann Sorge, dass die Datei menschenlesbar formatiert wird (Einrücken, Umbrüche, etc.). Des Weiteren wird auch der DocType in die Datei geschrieben. Die Methode setzt ausserdem die Breite und die Höhe, sowie die viewBox des SVG auf bessere Werte.

Parameter:

pathAndSvgName Der vollständige Pfad zur zu formatierenden Datei.

Definiert in Zeile 369 der Datei StaxUtil.java.

static void svgToolTipGroup (XMLEventWriter writer) [static]

Diese Methode schreibt einige Textzeilen ins SVG, welche für das Anzeigen der ToolTips nötig sind.

Parameter:

writer In diese Datei sollen die Zeilen geschrieben werden.

Definiert in Zeile 255 der Datei StaxUtil.java.

9.3.4.3 Dokumentation der Datenelemente

**final String linkEditInstanceObject =
"/../extendDataInput/show_ObjectInstance.jsf?" [static, private]**

Definiert in Zeile 31 der Datei StaxUtil.java.

Wird benutzt von StaxUtil.drawDataObject().

**final String linkEditInstanceRelation =
"/../extendDataInput/show_RelationInstance.jsf?" [static, private]**

Definiert in Zeile 35 der Datei StaxUtil.java.

Wird benutzt von StaxUtil.drawEdgeWithInfo().

final String linkMetaObjectId = "&metaobjectId=" [static, private]

Definiert in Zeile 29 der Datei StaxUtil.java.

Wird benutzt von StaxUtil.drawDataObject().

final String linkMetaRelationId = "&metarelationId=" [static, private]

Definiert in Zeile 33 der Datei StaxUtil.java.

Wird benutzt von StaxUtil.drawEdgeWithInfo().

final String linkObjectId = "&instanceOfMetaobject=" [static, private]

Definiert in Zeile 30 der Datei StaxUtil.java.

Wird benutzt von StaxUtil.drawDataObject().

final String linkRelationId = "&instanceOfMetaobject=" [static, private]

Definiert in Zeile 34 der Datei StaxUtil.java.

Wird benutzt von StaxUtil.drawEdgeWithInfo().

9.3.5 Util Klassenreferenz

9.3.5.1 Ausführliche Beschreibung

In dieser Klasse werden einige Hilfsmethoden gesammelt, die in diesem Bundle häufig verwendet werden.

Autor:

philipp

Definiert in Zeile 27 der Datei Util.java.

Öffentliche, statische Methoden

- static void copyFile (String src, String dest)
- static void deleteFilesFromDirectory (File directory)
- static void deleteOldSvgPngFiles (File directory)
- static void formatJspFile (String pathAndSvgName)
- static String getPathToModVis1 ()

Private, statische Methoden

- static void copy (InputStream in, OutputStream out) throws IOException

9.3.5.2 Dokumentation der Elementfunktionen

static void copy (InputStream in, OutputStream out) throws IOException [static, private]

Kopiert den InputStream in den OutputStream

Parameter:

in Quelle des Streams

out Ziel des Streams

Ausnahmebehandlung:**IOException**

Definiert in Zeile 88 der Datei Util.java.

Wird benutzt von Util.copyFile().

static void copyFile (String src, String dest) [static]

Kopiert mit Hilfe der Methode {

Siehe auch:

copy} die erste Datei am Ort src an den Ort dest.

Parameter:

src Quelldatei

dest Zieldatei

Definiert in Zeile 101 der Datei Util.java.

Benutzt Util.copy().

static void deleteFilesFromDirectory (File directory) [static]

Löscht alle Dateien im angegebenen Verzeichnis.

Parameter:

directory Das Verzeichnis

Definiert in Zeile 127 der Datei Util.java.

static void deleteOldSvgPngFiles (File directory) [static]

Löscht alle .svg und .png Dateien aus dem angegebenen Verzeichnis, die älter sind, als in {

Siehe auch:

Constants.OLD_FILE} angegeben (24h). Die Dateien, die gelöscht werden, fangen mit "_matirx" oder "_tree" an, oder sind SVGs.

Parameter:

directory Aus diesem Verzeichnis sollen alle "alten" Dateien entfernt werden.

Definiert in Zeile 146 der Datei Util.java.

static void formatJspFile (String pathAndSvgName) [static]

Diese Methode formatiert die übergebene Datei. Die Elemente werden eingerückt (geht mit allen xml-ähnlich strukturierten Dateien (svg, html, jsp, etc).

Parameter:

pathAndSvgName Diese Datei wird verändert

Definiert in Zeile 35 der Datei Util.java.

static String getPathToModVis1 () [static]

Diese Methode liefert den Pfad zum Verzeichnis zu de.offis.pg.eam.mod_vis_1 auf Grundlage dessen, was in der im Verzeichnis System.getProperty("user.dir") liegenden "config_eam_mod_vis.xml" steht.

Rückgabe:

Verzeichnispfad

Definiert in Zeile 64 der Datei Util.java.

9.4 Implementierung

9.4.1 Generator Klassenreferenz

9.4.1.1 Ausführliche Beschreibung

Diese Klasse ist zuständig für die Generierung der SVG-Bilder (Matrix / Baum).

Autor:

philipp

Definiert in Zeile 24 der Datei Generator.java.

Öffentliche, statische Methoden

- static void createSVGMatrixFromTree (IDataCollector treeCollector, String pathAndSvgName) throws Exception
- static void createSVGTreeFromTree (IDataCollector treeCollector, String pathAndSvgName) throws Exception
- static float getMaxHeight ()
- static float getMaxWidth ()
- static void setMaxHeight (float maxHeight)
- static void setMaxWidth (float maxWidth)

Private, statische Methoden

- static void createSVGTreeFromObjTree (XMLEventWriter writer, IDataCollector treeCollector)

- static void drawTreeRecursive (XMLEventWriter writer, float firstX, float firstY, float viewBoxWidth, float viewBoxHeight, float iconSize, TreeObject node)
- static String getQualifyingID (TreeObject object)

Statische private Attribute

- static HashMap< String, float[]> map = new HashMap<String, float[]>()
- static float maxHeight = 0f
- static float maxWidth = 0f

9.4.1.2 Dokumentation der Elementfunktionen

static void createSVGMatrixFromTree (IDataCollector treeCollector, String pathAndSvgName) throws Exception [static]

Erzeugt ein SVG-Bild. Es wird eine Verbindungsmatrix erzeugt, die einen Überblick über die Verbindungen der Objekte im übergebenen Baum gibt.

Parameter:

tree Die Objekte dieses Baumes werden in der Matrix auf der x- und y-Achse aufgetragen.

height Die Höhe des SVG

width Die Breite

pathAndSvgName Unter diesem Namen wird das SVG gespeichert, Bsp: Ordner/nochEinOrdner/BildName.svg

Ausnahmebehandlung:

Exception

Definiert in Zeile 46 der Datei Generator.java.

Benutzt Generator.map.

static void createSVGTreeFromObjTree (XMLEventWriter writer, IDataCollector treeCollector) [static, private]

An diese Methode wird ein Baum übergeben, dessen Wurzel ein Objekt und keine Relation ist.

Parameter:

writer Die Datei in die alles geschrieben wird

tree Der Baum, der gemalt werden soll

Definiert in Zeile 187 der Datei Generator.java.

Benutzt Generator.drawTreeRecursive(), Generator.maxHeight und Generator.maxWidth.

Wird benutzt von Generator.createSVGTreeFromTree().

static void createSVGTreeFromTree (IDataCollector treeCollector, String pathAndSvgName) throws Exception [static]

Diese Methode erzeugt an der angegebenen Adresse (pathAndSvgName) eine SVG-Grafik, die den übergebenen Baum darstellt.

Parameter:

tree Der Baum, der dargestellt werden soll.

height Die Höhe des SVG

width Die Breite des SVG

pathAndSvgName Unter diesem Namen wird das SVG gespeichert, Bsp:
Ordner/nochEinOrdner/BildName.svg

Ausnahmebehandlung:

Exception

Definiert in Zeile 150 der Datei Generator.java.

Benutzt Generator.createSVGTreeFromObjTree() und Generator.map.

static void drawTreeRecursive (XMLEventWriter writer, float firstX, float firstY, float viewBoxWidth, float viewBoxHeight, float iconSize, TreeObject node) [static, private]

Diese Methode erzeugt rekursiv die eigentliche Zeichnung des Baumes aus der Analyse.

Parameter:

writer In dieser Datei wird alles gespeichert

firstX hier soll die wurzel des baumes gemalt werden (x-Koordinate)

firstY hier soll die wurzel des baumes gemalt werden (y-Koordinate)

node Die Wurzel des Baumes, der gezeichnet werden soll.

Definiert in Zeile 262 der Datei Generator.java.

Benutzt Generator.getQualifyingID(), Generator.map, Generator.maxHeight und Generator.maxWidth.

Wird benutzt von Generator.createSVGTreeFromObjTree().

static float getMaxHeight () [static]

Definiert in Zeile 389 der Datei Generator.java.

Benutzt Generator.maxHeight.

static float getMaxWidth () [static]

Definiert in Zeile 379 der Datei Generator.java.

Benutzt Generator.maxWidth.

static String getQualifyingID (TreeObject object) [static, private]

Diese Methode berechnet aus dem Übergeben TreeObject eine ID, die systemweit einzigartig sein sollte. Sie hat die Form metaModellID_metaObjectID_ObjectID

Parameter:

object Ein TreeObject zu dessen DataObject die systemweit einzigartige ID berechnet werden soll

Rückgabe:

Die ID

Definiert in Zeile 243 der Datei Generator.java.

Wird benutzt von Generator.drawTreeRecursive().

static void setMaxHeight (float maxHeight) [static]

Definiert in Zeile 393 der Datei Generator.java.

static void setMaxWidth (float maxWidth) [static]

Definiert in Zeile 383 der Datei Generator.java.

9.4.1.3 Dokumentation der Datenelemente

HashMap<String, float[]> map = new HashMap<String, float[]>() [static, private]

Definiert in Zeile 27 der Datei Generator.java.

Wird benutzt von Generator.createSVGMatrixFromTree(), Generator.createSVGTreeFromTree() und Generator.drawTreeRecursive().

float maxHeight = 0f [static, private]

Definiert in Zeile 29 der Datei Generator.java.

Wird benutzt von Generator.createSVGTreeFromObjTree(), Generator.drawTreeRecursive() und Generator.getMaxHeight().

float maxWidth = 0f [static, private]

Definiert in Zeile 28 der Datei Generator.java.

Wird benutzt von Generator.createSVGTreeFromObjTree(), Generator.drawTreeRecursive() und Generator.getMaxWidth().

9.4.2 Relation Klassenreferenz

9.4.2.1 Ausführliche Beschreibung

Diese Klasse stellt Methoden zur Verfügung, mit denen die Matrix-Relation (Verbindungsmatrix) für die Visualisierung aus Bäumen der Analyse erzeugt werden kann.

Autor:

philipp

Definiert in Zeile 17 der Datei Relation.java.

Öffentliche, statische Methoden

- static String[][] computeMatrixFromTree (List< TreeObject > objects)
- static List< TreeObject > getObjectsFromTree (TreeObject tree)

Private, statische Methoden

- static String getQualifyingID (DataObject obj)
- static void listRelation (String[][] matrix)

9.4.2.2 Dokumentation der Elementfunktionen

static String [][] computeMatrixFromTree (List< TreeObject > objects) [static]

Erzeugt die Verbindungsmatrix zwischen den übergebenen TreeObjects

Parameter:

objects Die Liste der Objekte zu denen die Matrix erzeugt werden soll

Rückgabe:

Die Matrix

Definiert in Zeile 96 der Datei Relation.java.

Benutzt Relation.getQualifyingID().

static List<TreeObject> getObjectsFromTree (TreeObject tree) [static]

Liefert alle TreeObjects aus einem Baum

Parameter:

tree Der Baum, der übergeben wird.

Rückgabe:

Liste von TreeObjects

Definiert in Zeile 27 der Datei Relation.java.

static String getQualifyingID (DataObject obj) [static, private]

Diese Methode berechnet aus dem Übergeben TreeObject eine ID, die systemweit einzigartig sein sollte. Sie hat die Form metaModellID_metaObjectID_ObjectID

Parameter:

object Ein DataObject dessen systemweit einzigartige ID berechnet werden soll

Rückgabe:

Die ID

Definiert in Zeile 75 der Datei Relation.java.

Wird benutzt von Relation.computeMatrixFromTree().

static void listRelation (String matrix[][]) [static, private]

Gibt die eine quadratische Matrix (n x n) in der Console aus

Parameter:

matrix

Definiert in Zeile 155 der Datei Relation.java.

9.4.3 MatrixVisualization Klassenreferenz

9.4.3.1 Ausführliche Beschreibung

Diese klasse stellt Methoden zur Verfügung, die von den JSP-Seiten aus angesprochen werden können.

Autor:

philipp

Definiert in Zeile 30 der Datei MatrixVisualization.java.

Öffentliche Methoden

- String getCurrentUser ()
- String getHeadLineName ()
- String getSvgFilename ()
- String saveAndStoreInContainer ()

- void setCurrentUser (String currentUser)

Öffentliche, statische Methoden

- static String drawNewSVG ()
- static DataObject getInstance ()
- static HashMap< String, String > getMap ()
- static float getSvgHeight ()
- static String getSvgName ()
- static float getSvgWidth ()
- static void setHeadLineName (String name)
- static void setInstance (DataObject object)
- static void setMap (HashMap< String, String > map)
- static void setSvgHeight (float svgHeight)
- static void setSvgName (String name)
- static void setSvgWidth (float svgWidth)

Private Methoden

- String getPngName ()

Private Attribute

- String currentUser = (String) FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Statische private Attribute

- static String headLineName = ""
- static DataObject instance = null
- static HashMap< String, String > map = null
- static String png = Constants.PNG_FOLDER_NAME
- static String sep = Constants.SEP
- static String svg = Constants.SVG_FOLDER_NAME
- static float svgHeight = 0f
- static String svgName = ""
- static float svgWidth = 0f

9.4.3.2 Dokumentation der Elementfunktionen

static String drawNewSVG () [static]

Diese Methode generiert das SVG-Matrix-Bild

Rückgabe:

”done” So wird festgestellt, wann die Methode fertig ist.

Definiert in Zeile 55 der Datei MatrixVisualization.java.

Benutzt AnalysisAdapter.getCollector(), MatrixVisualization.getSvgName(), MatrixVisualization.instance, MatrixVisualization.map, MatrixVisualization.sep und MatrixVisualization.svg.

String getCurrentUser ()

Definiert in Zeile 144 der Datei MatrixVisualization.java.

Benutzt MatrixVisualization.currentUser.

Wird benutzt von MatrixVisualization.saveAndStoreInContainer().

String getHeadLineName ()

Definiert in Zeile 211 der Datei MatrixVisualization.java.

Benutzt MatrixVisualization.headLineName.

static DataObject getInstance () [static]

Definiert in Zeile 164 der Datei MatrixVisualization.java.

Benutzt MatrixVisualization.instance.

static HashMap<String, String> getMap () [static]

Definiert in Zeile 197 der Datei MatrixVisualization.java.

Benutzt MatrixVisualization.map.

String getPngName () [private]

Definiert in Zeile 154 der Datei MatrixVisualization.java.

Benutzt MatrixVisualization.instance.

Wird benutzt von MatrixVisualization.saveAndStoreInContainer().

String getSvgFilename ()

Definiert in Zeile 170 der Datei MatrixVisualization.java.

static float getSvgHeight () [static]

Definiert in Zeile 187 der Datei MatrixVisualization.java.

Benutzt MatrixVisualization.svgHeight.

static String getSvgName () [static]

Definiert in Zeile 134 der Datei MatrixVisualization.java.

Benutzt MatrixVisualization.svgName.

Wird benutzt von MatrixVisualization.drawNewSVG() und MatrixVisualization.saveAndStoreInContainer().

static float getSvgWidth () [static]

Definiert in Zeile 177 der Datei MatrixVisualization.java.

Benutzt MatrixVisualization.svgWidth.

String saveAndStoreInContainer ()

Diese Methode speichert das generierte SVG als PNG-Datei und legt es dann für den Benutzer in den Container. Dort kann es vom Export-Modul entnommen werden.

Rückgabe:

 null Es ist keine für die Navigation relevante Rückgabe nötig.

Definiert in Zeile 90 der Datei MatrixVisualization.java.

Benutzt MatrixVisualization.getCurrentUser(), MatrixVisualization.getPngName(), MatrixVisualization.getSvgName(), MatrixVisualization.png, MatrixVisualization.sep, MatrixVisualization.setCurrentUser() und MatrixVisualization.svg.

void setCurrentUser (String currentUser)

Definiert in Zeile 148 der Datei MatrixVisualization.java.

Wird benutzt von MatrixVisualization.saveAndStoreInContainer().

static void setHeadLineName (String name) [static]

Definiert in Zeile 207 der Datei MatrixVisualization.java.

Benutzt MatrixVisualization.headLineName.

static void setInstance (DataObject object) [static]

Definiert in Zeile 160 der Datei MatrixVisualization.java.

Benutzt MatrixVisualization.instance.

static void setMap (HashMap< String, String > map) [static]

Definiert in Zeile 201 der Datei MatrixVisualization.java.

static void setSvgHeight (float svgHeight) [static]

Definiert in Zeile 191 der Datei MatrixVisualization.java.

static void setSvgName (String name) [static]

Definiert in Zeile 138 der Datei MatrixVisualization.java.

Benutzt MatrixVisualization.svgName.

static void setSvgWidth (float svgWidth) [static]

Definiert in Zeile 181 der Datei MatrixVisualization.java.

9.4.3.3 Dokumentation der Datenelemente

String currentUser = (String)

FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")
[private]

Definiert in Zeile 44 der Datei MatrixVisualization.java.

Wird benutzt von MatrixVisualization.getCurrentUser().

String headLineName = "" [static, private]

Definiert in Zeile 41 der Datei MatrixVisualization.java.

Wird benutzt von MatrixVisualization.getHeadLineName() und
MatrixVisualization.setHeadLineName().

DataObject instance = null [static, private]

Definiert in Zeile 32 der Datei MatrixVisualization.java.

Wird benutzt von MatrixVisualization.drawNewSVG(), MatrixVisualization.getInstance(),
MatrixVisualization.getPngName() und MatrixVisualization.setInstance().

HashMap<String, String> map = null [static, private]

Definiert in Zeile 34 der Datei MatrixVisualization.java.

Wird benutzt von MatrixVisualization.drawNewSVG() und MatrixVisualization.getMap().

String png = Constants.PNG_FOLDER_NAME [static, private]

Definiert in Zeile 48 der Datei MatrixVisualization.java.

Wird benutzt von MatrixVisualization.saveAndStoreInContainer().

String sep = Constants.SEP [static, private]

Definiert in Zeile 46 der Datei MatrixVisualization.java.

Wird benutzt von MatrixVisualization.drawNewSVG() und MatrixVisualization.saveAndStoreInContainer().

String svg = Constants.SVG_FOLDER_NAME [static, private]

Definiert in Zeile 47 der Datei MatrixVisualization.java.

Wird benutzt von MatrixVisualization.drawNewSVG() und MatrixVisualization.saveAndStoreInContainer().

float svgHeight = 0f [static, private]

Definiert in Zeile 39 der Datei MatrixVisualization.java.

Wird benutzt von MatrixVisualization.getSvgHeight().

String svgName = "" [static, private]

Definiert in Zeile 36 der Datei MatrixVisualization.java.

Wird benutzt von MatrixVisualization.getSvgName() und MatrixVisualization.setSvgName().

float svgWidth = 0f [static, private]

Definiert in Zeile 38 der Datei MatrixVisualization.java.

Wird benutzt von MatrixVisualization.getSvgWidth().

9.4.4 SelectObject Klassenreferenz

9.4.4.1 Ausführliche Beschreibung

Diese Klasse nimmt die Eingaben des Nutzers entgegen.

Autor:

philipp

Definiert in Zeile 39 der Datei SelectObject.java.

Öffentliche Methoden

- String analyse ()
- String getCostTypeObject ()
- HashMap< String, String > getMap ()
- String getPerson ()
- String getRelationPerson ()
- String getSelectedDisplaystyle ()
- String getSelectedInstance ()
- String getSelectedMetamodel ()
- String getSelectedObject ()
- LinkedList< SelectItem > getSelectOneOfTheseDisplaystyles ()
- LinkedList< SelectItem > getSelectOneOfTheseInstances ()
- LinkedList< SelectItem > getSelectOneOfTheseMetamodels ()
- LinkedList< SelectItem > getSelectOneOfTheseObjects ()
- boolean isMatrix ()
- boolean isTree ()
- void selectDisplaystyle (ValueChangeEvent valueChangeEvent)
- void selectInstance (ValueChangeEvent valueChangeEvent)
- void selectMetamodel (ValueChangeEvent valueChangeEvent)
- void selectObject (ValueChangeEvent valueChangeEvent)
- SelectObject ()
- void setCostTypeObject (String cost)
- void setMap (HashMap< String, String > map)
- void setMatrix (boolean matrix)
- void setPerson (String person)
- void setRelationPerson (String relationPerson)
- void setSelectedDisplaystyle (String selectedDisplaystyle)
- void setSelectedInstance (String instance)
- void setSelectedMetamodel (String metamodel)
- void setSelectedObject (String object)
- void setTree (boolean tree)

Private Methoden

- void initInstances ()
- void initMetamodels ()
- void initObjects ()
- String readTemplate (String path)

Private Attribute

- final EAMCategoryDAO categoryDAO = DAOFactory.getAuthInstance(userId, password).getEAMCategoryDAO()
- String costTypeObject = "server"
- final EAMObjectDAO eamObjectDAO = DAOFactory.getAuthInstance(userId, password).getEAMObjectDAO()
- final EAMRelationDAO eamRelationDAO = DAOFactory.getAuthInstance(userId, password).getEAMRelationDAO()
- DataObject instance = null
- HashMap< String, String > map = new HashMap<String, String>()
- boolean matrix = false
- EAMCategory metamodel = null
- EAMSuperClass object = null
- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String person = "verantwortlicher"
- String relationPerson = "hat"
- String selectedDisplaystyle = null
- String selectedInstance = null
- String selectedMetamodel = null
- String selectedObject = null
- LinkedList< SelectItem > selectOneOfTheseDisplaystyles = new
LinkedList<SelectItem>()
- LinkedList< SelectItem > selectOneOfTheseInstances = new
LinkedList<SelectItem>()
- LinkedList< SelectItem > selectOneOfTheseMetamodels = new
LinkedList<SelectItem>()
- LinkedList< SelectItem > selectOneOfTheseObjects = new
LinkedList<SelectItem>()
- boolean tree = false
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

9.4.4.2 Beschreibung der Konstruktoren und Destruktoren**SelectObject ()**

Definiert in Zeile 78 der Datei SelectObject.java.

Benutzt SelectObject.selectOneOfTheseDisplaystyles.

9.4.4.3 Dokumentation der Elementfunktionen

String analyse ()

Diese Methode wird aufgerufen, wenn der Nutzer auf den "Analyse >>" -Knopf auf der JSP-Seite drückt.

Rückgabe:

String für die Navigation: mod1ObjectSuccessfullySelectedDrawMatrix oder mod1ObjectSuccessfullySelectedDrawTree, falls erfolgreich, null sonst.

Definiert in Zeile 361 der Datei SelectObject.java.

Benutzt SelectObject.costTypeObject, SelectObject.instance, SelectObject.isMatrix(), SelectObject.isTree(), SelectObject.map, SelectObject.person, SelectObject.readTemplate() und SelectObject.relationPerson.

String getCostTypeObject ()

Definiert in Zeile 507 der Datei SelectObject.java.

Benutzt SelectObject.costTypeObject.

HashMap<String, String> getMap ()

Definiert in Zeile 489 der Datei SelectObject.java.

Benutzt SelectObject.map.

String getPerson ()

Definiert in Zeile 495 der Datei SelectObject.java.

Benutzt SelectObject.person.

String getRelationPerson ()

Definiert in Zeile 501 der Datei SelectObject.java.

Benutzt SelectObject.relationPerson.

String getSelectedDisplaystyle ()

Definiert in Zeile 482 der Datei SelectObject.java.

Benutzt SelectObject.selectedDisplaystyle.

String getSelectedInstance ()

Definiert in Zeile 232 der Datei SelectObject.java.

Benutzt SelectObject.selectedInstance.

String getSelectedMetamodel ()

Definiert in Zeile 88 der Datei SelectObject.java.

Benutzt SelectObject.selectedMetamodel.

String getSelectedObject ()

Definiert in Zeile 146 der Datei SelectObject.java.

Benutzt SelectObject.selectedObject.

LinkedList<SelectItem> getSelectOneOfTheseDisplaystyles ()

Definiert in Zeile 346 der Datei SelectObject.java.

Benutzt SelectObject.selectOneOfTheseDisplaystyles.

LinkedList<SelectItem> getSelectOneOfTheseInstances ()

Definiert in Zeile 325 der Datei SelectObject.java.

Benutzt SelectObject.selectOneOfTheseInstances.

LinkedList<SelectItem> getSelectOneOfTheseMetamodels ()

Definiert in Zeile 119 der Datei SelectObject.java.

Benutzt SelectObject.initMetamodels() und SelectObject.selectOneOfTheseMetamodels.

LinkedList<SelectItem> getSelectOneOfTheseObjects ()

Definiert in Zeile 212 der Datei SelectObject.java.

Benutzt SelectObject.selectOneOfTheseObjects.

void initInstances () [private]

Definiert in Zeile 242 der Datei SelectObject.java.

Benutzt SelectObject.password, SelectObject.selectOneOfTheseInstances und SelectObject.userId.

Wird benutzt von SelectObject.setSelectedObject().

void initMetamodels () [private]

Definiert in Zeile 127 der Datei SelectObject.java.

Benutzt SelectObject.categoryDAO und SelectObject.selectOneOfTheseMetamodels.

Wird benutzt von SelectObject.getSelectOneOfTheseMetamodels().

void initObjects () [private]

Definiert in Zeile 160 der Datei SelectObject.java.

Benutzt SelectObject.eamObjectDAO, SelectObject.metamodel und SelectObject.selectOneOfTheseObjects.

Wird benutzt von SelectObject.setSelectedMetamodel().

boolean isMatrix ()

Definiert in Zeile 469 der Datei SelectObject.java.

Benutzt SelectObject.matrix.

Wird benutzt von SelectObject.analyse().

boolean isTree ()

Definiert in Zeile 475 der Datei SelectObject.java.

Benutzt SelectObject.tree.

Wird benutzt von SelectObject.analyse().

String readTemplate (String path) [private]

Diese Methode liest die Datei unter dem gegebenen Pfad ein.

Parameter:

path Hier liegt die Textdatei.

Rückgabe:

Ein String, der den Inhalt der Textdatei enthält.

Definiert in Zeile 443 der Datei SelectObject.java.

Wird benutzt von SelectObject.analyse().

void selectDisplayStyle (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 332 der Datei SelectObject.java.

Benutzt SelectObject.selectedDisplayStyle, SelectObject.setMatrix(), SelectObject.setSelectedDisplayStyle() und SelectObject.setTree().

void selectInstance (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 277 der Datei SelectObject.java.

Benutzt SelectObject.instance, SelectObject.password, SelectObject.selectedInstance, SelectObject.setSelectedInstance() und SelectObject.userId.

void selectMetamodel (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 109 der Datei SelectObject.java.

Benutzt SelectObject.categoryDAO, SelectObject.metamodel, SelectObject.selectedMetamodel und SelectObject.setSelectedMetamodel().

void selectObject (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 188 der Datei SelectObject.java.

Benutzt SelectObject.eamObjectDAO, SelectObject.eamRelationDAO, SelectObject.selectedObject und SelectObject.setSelectedObject().

void setCostTypeObject (String cost)

Definiert in Zeile 510 der Datei SelectObject.java.

Benutzt SelectObject.costTypeObject.

void setMap (HashMap< String, String > map)

Definiert in Zeile 492 der Datei SelectObject.java.

void setMatrix (boolean matrix)

Definiert in Zeile 472 der Datei SelectObject.java.

Wird benutzt von SelectObject.selectDisplaystyle().

void setPerson (String person)

Definiert in Zeile 498 der Datei SelectObject.java.

void setRelationPerson (String relationPerson)

Definiert in Zeile 504 der Datei SelectObject.java.

void setSelectedDisplaystyle (String selectedDisplaystyle)

Definiert in Zeile 485 der Datei SelectObject.java.

Wird benutzt von SelectObject.selectDisplaystyle().

void setSelectedInstance (String instance)

Definiert in Zeile 236 der Datei SelectObject.java.

Benutzt SelectObject.selectedInstance.

Wird benutzt von SelectObject.selectInstance().

void setSelectedMetamodel (String metamodel)

Definiert in Zeile 91 der Datei SelectObject.java.

Benutzt SelectObject.initObjects(), SelectObject.instance, SelectObject.selectedInstance, SelectObject.selectedMetamodel, SelectObject.selectedObject, SelectObject.selectOneOfTheseInstances und SelectObject.selectOneOfTheseObjects.

Wird benutzt von SelectObject.selectMetamodel().

void setSelectedObject (String object)

Definiert in Zeile 150 der Datei SelectObject.java.

Benutzt SelectObject.initInstances() und SelectObject.selectedObject.

Wird benutzt von SelectObject.selectObject().

void setTree (boolean tree)

Definiert in Zeile 478 der Datei SelectObject.java.

Wird benutzt von SelectObject.selectDisplaystyle().

9.4.4.4 Dokumentation der Datenelemente

final EAMCategoryDAO categoryDAO = DAOFactory.getAuthInstance(userId, password).getEAMCategoryDAO() [private]

Definiert in Zeile 46 der Datei SelectObject.java.

Wird benutzt von SelectObject.initMetamodels() und SelectObject.selectMetamodel().

String costTypeObject = "server" [private]

Definiert in Zeile 72 der Datei SelectObject.java.

Wird benutzt von SelectObject.analyse(), SelectObject.getCostTypeObject() und SelectObject.setCostTypeObject().

final EAMObjectDAO eamObjectDAO = DAOFactory.getAuthInstance(userId, password).getEAMObjectDAO() [private]

Definiert in Zeile 47 der Datei SelectObject.java.

Wird benutzt von SelectObject.initObjects() und SelectObject.selectObject().

final EAMRelationDAO eamRelationDAO = DAOFactory.getAuthInstance(userId, password).getEAMRelationDAO() [private]

Definiert in Zeile 48 der Datei SelectObject.java.

Wird benutzt von `SelectObject.selectObject()`.

DataObject instance = null [private]

Definiert in Zeile 59 der Datei `SelectObject.java`.

Wird benutzt von `SelectObject.analyse()`, `SelectObject.selectInstance()` und `SelectObject.setSelectedMetamodel()`.

HashMap<String, String> map = new HashMap<String, String>() [private]

Definiert in Zeile 69 der Datei `SelectObject.java`.

Wird benutzt von `SelectObject.analyse()` und `SelectObject.getMap()`.

boolean matrix = false [private]

Definiert in Zeile 66 der Datei `SelectObject.java`.

Wird benutzt von `SelectObject.isMatrix()`.

EAMCategory metamodel = null [private]

Definiert in Zeile 51 der Datei `SelectObject.java`.

Wird benutzt von `SelectObject.initObjects()` und `SelectObject.selectMetamodel()`.

EAMSuperClass object = null [private]

Definiert in Zeile 55 der Datei `SelectObject.java`.

String password =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
[private]

Definiert in Zeile 43 der Datei `SelectObject.java`.

Wird benutzt von `SelectObject.initInstances()` und `SelectObject.selectInstance()`.

String person = "verantwortlicher" [private]

Definiert in Zeile 70 der Datei `SelectObject.java`.

Wird benutzt von `SelectObject.analyse()` und `SelectObject.getPerson()`.

String relationPerson = "hat" [private]

Definiert in Zeile 71 der Datei `SelectObject.java`.

Wird benutzt von `SelectObject.analyse()` und `SelectObject.getRelationPerson()`.

String selectedDisplaystyle = null [private]

Definiert in Zeile 63 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedDisplaystyle() und SelectObject.selectDisplaystyle().

String selectedInstance = null [private]

Definiert in Zeile 60 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedInstance(), SelectObject.selectInstance(), SelectObject.setSelectedInstance() und SelectObject.setSelectedMetamodel().

String selectedMetamodel = null [private]

Definiert in Zeile 52 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedMetamodel(), SelectObject.selectMetamodel() und SelectObject.setSelectedMetamodel().

String selectedObject = null [private]

Definiert in Zeile 56 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedObject(), SelectObject.selectObject(), SelectObject.setSelectedMetamodel() und SelectObject.setSelectedObject().

**LinkedList<SelectItem> selectOneOfTheseDisplaystyles = new
LinkedList<SelectItem>() [private]**

Definiert in Zeile 64 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectOneOfTheseDisplaystyles() und SelectObject.SelectObject().

**LinkedList<SelectItem> selectOneOfTheseInstances = new
LinkedList<SelectItem>() [private]**

Definiert in Zeile 61 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectOneOfTheseInstances(), SelectObject.initInstances() und SelectObject.setSelectedMetamodel().

**LinkedList<SelectItem> selectOneOfTheseMetamodels = new
LinkedList<SelectItem>() [private]**

Definiert in Zeile 53 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectOneOfTheseMetamodels() und SelectObject.initMetamodels().

```
LinkedList<SelectItem> selectOneOfTheseObjects = new  
LinkedList<SelectItem>() [private]
```

Definiert in Zeile 57 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedOneOfTheseObjects(), SelectObject.initObjects() und SelectObject.setSelectedMetamodel().

```
boolean tree = false [private]
```

Definiert in Zeile 67 der Datei SelectObject.java.

Wird benutzt von SelectObject.isTree().

```
String userId =  
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")  
[private]
```

Definiert in Zeile 42 der Datei SelectObject.java.

Wird benutzt von SelectObject.initInstances() und SelectObject.selectInstance().

9.4.5 TreeVisualization Klassenreferenz

9.4.5.1 Ausführliche Beschreibung

Diese Klasse agiert als Bean für die Darstellung der SVGs, die die Verbindungsbäume enthalten. Hier finden sich die Methoden, die von den entsprechenden .jsp-Seiten aus aufgerufen werden.

Autor:

philipp

Definiert in Zeile 29 der Datei TreeVisualization.java.

Öffentliche Methoden

- String getCurrentUser ()
- String getHeadLineName ()
- String getSvgFilename ()
- String saveAndStoreInContainer ()
- void setCurrentUser (String currentUser)

Öffentliche, statische Methoden

- static String drawNewSVG ()

- static DataObject getInstance ()
- static HashMap< String, String > getMap ()
- static float getSvgHeight ()
- static String getSvgName ()
- static float getSvgWidth ()
- static void setHeadLineName (String name)
- static void setInstance (DataObject object)
- static void setMap (HashMap< String, String > map)
- static void setSvgHeight (float svgHeight)
- static void setSvgName (String name)
- static void setSvgWidth (float svgWidth)

Private Methoden

- String getPngName ()

Private Attribute

- String currentUser = (String) FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

Statische private Attribute

- static String headLineName = ""
- static DataObject instance = null
- static HashMap< String, String > map = null
- static String png = Constants.PNG_FOLDER_NAME
- static String sep = Constants.SEP
- static String svg = Constants.SVG_FOLDER_NAME
- static float svgHeight = 0f
- static String svgName = ""
- static float svgWidth = 0f

9.4.5.2 Dokumentation der Elementfunktionen

static String drawNewSVG () [static]

Diese Methode generiert das SVG-Tree-Bild

Rückgabe:

"done" So wird festgestellt, wann die Methode fertig ist.

Definiert in Zeile 58 der Datei TreeVisualization.java.

Benutzt AnalysisAdapter.getCollector(), TreeVisualization.getSvgName(), TreeVisualization.instance, TreeVisualization.map, TreeVisualization.sep und TreeVisualization.svg.

String getCurrentUser ()

Definiert in Zeile 154 der Datei TreeVisualization.java.

Benutzt TreeVisualization.currentUser.

Wird benutzt von TreeVisualization.saveAndStoreInContainer().

String getHeadLineName ()

Definiert in Zeile 139 der Datei TreeVisualization.java.

Benutzt TreeVisualization.headLineName.

static DataObject getInstance () [static]

Definiert in Zeile 173 der Datei TreeVisualization.java.

Benutzt TreeVisualization.instance.

static HashMap<String, String> getMap () [static]

Definiert in Zeile 206 der Datei TreeVisualization.java.

Benutzt TreeVisualization.map.

String getPngName () [private]

Definiert in Zeile 164 der Datei TreeVisualization.java.

Benutzt TreeVisualization.instance.

Wird benutzt von TreeVisualization.saveAndStoreInContainer().

String getSvgFilename ()

Definiert in Zeile 179 der Datei TreeVisualization.java.

static float getSvgHeight () [static]

Definiert in Zeile 196 der Datei TreeVisualization.java.

Benutzt TreeVisualization.svgHeight.

static String getSvgName () [static]

Definiert in Zeile 145 der Datei TreeVisualization.java.

Benutzt TreeVisualization.svgName.

Wird benutzt von TreeVisualization.drawNewSVG() und TreeVisualization.saveAndStoreInContainer().

static float getSvgWidth () [static]

Definiert in Zeile 186 der Datei TreeVisualization.java.

Benutzt TreeVisualization.svgWidth.

String saveAndStoreInContainer ()

Diese Methode speichert das generierte SVG als PNG-Datei und legt es dann für den Benutzer in den Container. Dort kann es vom Export-Modul entnommen werden.

Rückgabe:

 null Es ist keine für die Navigation relevante Rückgabe nötig.

Definiert in Zeile 93 der Datei TreeVisualization.java.

Benutzt TreeVisualization.getCurrentUser(), TreeVisualization.getPngName(), TreeVisualization.getSvgName(), TreeVisualization.png, TreeVisualization.sep, TreeVisualization.setCurrentUser() und TreeVisualization.svg.

void setCurrentUser (String currentUser)

Definiert in Zeile 158 der Datei TreeVisualization.java.

Wird benutzt von TreeVisualization.saveAndStoreInContainer().

static void setHeadLineName (String name) [static]

Definiert in Zeile 135 der Datei TreeVisualization.java.

Benutzt TreeVisualization.headLineName.

static void setInstance (DataObject object) [static]

Definiert in Zeile 170 der Datei TreeVisualization.java.

Benutzt TreeVisualization.instance.

static void setMap (HashMap< String, String > map) [static]

Definiert in Zeile 210 der Datei TreeVisualization.java.

static void setSvgHeight (float svgHeight) [static]

Definiert in Zeile 200 der Datei TreeVisualization.java.

static void setSvgName (String name) [static]

Definiert in Zeile 148 der Datei TreeVisualization.java.

Benutzt TreeVisualization.svgName.

static void setSvgWidth (float svgWidth) [static]

Definiert in Zeile 190 der Datei TreeVisualization.java.

9.4.5.3 Dokumentation der Datenelemente

String currentUser = (String)

FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")
[private]

Definiert in Zeile 42 der Datei TreeVisualization.java.

Wird benutzt von TreeVisualization.getCurrentUser().

String headLineName = "" [static, private]

Definiert in Zeile 39 der Datei TreeVisualization.java.

Wird benutzt von TreeVisualization.getHeadLineName() und
TreeVisualization.setHeadLineName().

DataObject instance = null [static, private]

Definiert in Zeile 31 der Datei TreeVisualization.java.

Wird benutzt von TreeVisualization.drawNewSVG(), TreeVisualization.getInstance(),
TreeVisualization.getPngName() und TreeVisualization.setInstance().

HashMap<String, String> map = null [static, private]

Definiert in Zeile 32 der Datei TreeVisualization.java.

Wird benutzt von TreeVisualization.drawNewSVG() und TreeVisualization.getMap().

String png = Constants.PNG_FOLDER_NAME [static, private]

Definiert in Zeile 46 der Datei TreeVisualization.java.

Wird benutzt von TreeVisualization.saveAndStoreInContainer().

String sep = Constants.SEP [static, private]

Definiert in Zeile 44 der Datei TreeVisualization.java.

Wird benutzt von TreeVisualization.drawNewSVG() und TreeVisualization.saveAndStoreInContainer().

String svg = Constants.SVG_FOLDER_NAME [static, private]

Definiert in Zeile 45 der Datei TreeVisualization.java.

Wird benutzt von TreeVisualization.drawNewSVG() und TreeVisualization.saveAndStoreInContainer().

float svgHeight = 0f [static, private]

Definiert in Zeile 37 der Datei TreeVisualization.java.

Wird benutzt von TreeVisualization.getSvgHeight().

String svgName = "" [static, private]

Definiert in Zeile 34 der Datei TreeVisualization.java.

Wird benutzt von TreeVisualization.getSvgName() und TreeVisualization.setSvgName().

float svgWidth = 0f [static, private]

Definiert in Zeile 36 der Datei TreeVisualization.java.

Wird benutzt von TreeVisualization.getSvgWidth().

10 de.offis.pg.eam.mod_vis_2

10.1 BundleConfiguration

10.1.1 BundleViews Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::module::api::IModuleView.

10.1.1.1 Ausführliche Beschreibung

Definiert in Zeile 22 der Datei BundleViews.java.

Öffentliche Methoden

- String getBundleVersion ()
- String getDescription ()
- long getEquinoxBundleId ()
- String getName ()
- List< View > getViews ()

Statische private Attribute

- static final String VIEWS = "bundle_views.xml"

10.1.1.2 Dokumentation der Elementfunktionen

String getBundleVersion ()

Definiert in Zeile 84 der Datei BundleViews.java.

String getDescription ()

Definiert in Zeile 29 der Datei BundleViews.java.

long getEquinoxBundleId ()

Definiert in Zeile 35 der Datei BundleViews.java.

String getName ()

Definiert in Zeile 41 der Datei BundleViews.java.

List<View> getViews ()

Definiert in Zeile 47 der Datei BundleViews.java.

Benutzt BundleViews.VIEWS.

10.1.1.3 Dokumentation der Datenelemente

final String VIEWS = "bundle_views.xml" [static, private]

Definiert in Zeile 25 der Datei BundleViews.java.

Wird benutzt von BundleViews.getViews().

10.1.2 Configuration Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::configuration::api::IBundleConfiguration.

10.1.2.1 Ausführliche Beschreibung

Definiert in Zeile 10 der Datei Configuration.java.

Öffentliche Methoden

- boolean bundleHasObjects ()
- boolean bundleHasRelations ()
- boolean bundleHasViews ()
- Document getObjectsJdomDocument () throws JDOMException, IOException

10.1.2.2 Dokumentation der Elementfunktionen

boolean bundleHasObjects ()

Definiert in Zeile 13 der Datei Configuration.java.

boolean bundleHasRelations ()

Definiert in Zeile 28 der Datei Configuration.java.

boolean bundleHasViews ()

Definiert in Zeile 18 der Datei Configuration.java.

Document `getObjectsJdomDocument ()` throws `JDOMException`, `IOException`

Definiert in Zeile 23 der Datei `Configuration.java`.

10.1.3 Activator Klassenreferenz

Abgeleitet von `org::osgi::framework::BundleActivator`.

10.1.3.1 Ausführliche Beschreibung

Definiert in Zeile 31 der Datei `Activator.java`.

Öffentliche Methoden

- `void start (BundleContext context) throws Exception`
- `void stop (BundleContext context) throws Exception`

Statische öffentliche Attribute

- `static String bundleSymbolicName`
- `static String bundleVersion`
- `static long equinoxBundleId`

Private Attribute

- `ServiceTracker httpServiceTracker`

Klassen

- `class HttpServiceTracker`
- `class ServletContextListenerServletAdaptor`

10.1.3.2 Dokumentation der Elementfunktionen**`void start (BundleContext context) throws Exception`**

Definiert in Zeile 39 der Datei `Activator.java`.

Benutzt `Activator.bundleSymbolicName`, `Activator.bundleVersion`, `Activator.equinoxBundleId` und `Activator.httpServiceTracker`.

void stop (BundleContext context) throws Exception

Definiert in Zeile 49 der Datei Activator.java.

Benutzt Activator.httpServiceTracker.

10.1.3.3 Dokumentation der Datenelemente

String bundleSymbolicName [static]

Definiert in Zeile 35 der Datei Activator.java.

Wird benutzt von Activator.start().

String bundleVersion [static]

Definiert in Zeile 36 der Datei Activator.java.

Wird benutzt von Activator.start().

long equinoxBundleId [static]

Definiert in Zeile 37 der Datei Activator.java.

Wird benutzt von Activator.start().

ServiceTracker httpServiceTracker [private]

Definiert in Zeile 33 der Datei Activator.java.

Wird benutzt von Activator.start() und Activator.stop().

10.1.4 Activator.HttpServiceTracker Klassenreferenz

Abgeleitet von org::osgi::util::tracker::ServiceTracker.

10.1.4.1 Ausführliche Beschreibung

Definiert in Zeile 54 der Datei Activator.java.

Öffentliche Methoden

- Object addingService (ServiceReference reference)
- HttpServiceTracker (BundleContext context)
- void removedService (ServiceReference reference, Object service)

Statische private Attribute

- static final String PATH = "/" + Constants.MODULE_NAME
- static final String WEB_ROOT = "/WebRoot"

10.1.4.2 Beschreibung der Konstruktoren und Destruktoren

HttpServiceTracker (BundleContext context)

Definiert in Zeile 59 der Datei Activator.java.

10.1.4.3 Dokumentation der Elementfunktionen

Object addingService (ServiceReference reference)

Definiert in Zeile 63 der Datei Activator.java.

Benutzt Activator.HttpServiceTracker.PATH und Activator.HttpServiceTracker.WEB_ROOT.

void removedService (ServiceReference reference, Object service)

Definiert in Zeile 99 der Datei Activator.java.

Benutzt Activator.HttpServiceTracker.PATH.

10.1.4.4 Dokumentation der Datenelemente

final String PATH = "/" + Constants.MODULE_NAME [static, private]

Definiert in Zeile 56 der Datei Activator.java.

Wird benutzt von Activator.HttpServiceTracker.addingService() und Activator.HttpServiceTracker.removedService().

final String WEB_ROOT = "/WebRoot" [static, private]

Definiert in Zeile 57 der Datei Activator.java.

Wird benutzt von Activator.HttpServiceTracker.addingService().

10.1.5 Activator.ServletContextListenerServletAdaptor Klassenreferenz

Abgeleitet von javax::servlet::Servlet.

10.1.5.1 Ausführliche Beschreibung

Definiert in Zeile 108 der Datei Activator.java.

Öffentliche Methoden

- void destroy ()
- ServletConfig getServletConfig ()
- String getServletInfo ()
- void init (ServletConfig config) throws ServletException
- void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException
- ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Private Attribute

- ServletConfig config
- Servlet delegate
- ClassLoader jspLoader
- ServletContextListener listener

10.1.5.2 Beschreibung der Konstruktoren und Destruktoren

ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Definiert in Zeile 114 der Datei Activator.java.

10.1.5.3 Dokumentation der Elementfunktionen

void destroy ()

Definiert in Zeile 148 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config, Activator.ServletContextListenerServletAdaptor.delegate, Activator.ServletContextListenerServletAdaptor.jspLoader und Activator.ServletContextListenerServletAdaptor.listener.

ServletConfig getServletConfig ()

Definiert in Zeile 162 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config.

String getServletInfo ()

Definiert in Zeile 166 der Datei Activator.java.

void init (ServletConfig config) throws ServletException

Definiert in Zeile 122 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config,
Activator.ServletContextListenerServletAdaptor.delegate,
Activator.ServletContextListenerServletAdaptor.jspLoader und
Activator.ServletContextListenerServletAdaptor.listener.

void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException

Definiert in Zeile 136 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.delegate und
Activator.ServletContextListenerServletAdaptor.jspLoader.

10.1.5.4 Dokumentation der Datenelemente**ServletConfig config [private]**

Definiert in Zeile 109 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.getServletConfig() und
Activator.ServletContextListenerServletAdaptor.init().

Servlet delegate [private]

Definiert in Zeile 111 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.init() und
Activator.ServletContextListenerServletAdaptor.service().

ClassLoader jspLoader [private]

Definiert in Zeile 112 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.init() und
Activator.ServletContextListenerServletAdaptor.service().

ServletContextListener listener [private]

Definiert in Zeile 110 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy() und
Activator.ServletContextListenerServletAdaptor.init().

10.1.6 ModVis2MenuContributor Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::api::AbstractAdoptableMenuContributor.

10.1.6.1 Ausführliche Beschreibung

Definiert in Zeile 6 der Datei ModVis2MenuContributor.java.

Öffentliche Methoden

- String getAdopterId ()
- String getContributionTarget ()
- String getContributorId ()
- String getTargetSubmenu ()
- ModVis2MenuContributor ()

10.1.6.2 Beschreibung der Konstruktoren und Destruktoren

ModVis2MenuContributor ()

Definiert in Zeile 8 der Datei ModVis2MenuContributor.java.

10.1.6.3 Dokumentation der Elementfunktionen

String getAdopterId ()

Definiert in Zeile 12 der Datei ModVis2MenuContributor.java.

String getContributionTarget ()

Definiert in Zeile 16 der Datei ModVis2MenuContributor.java.

String getContributorId ()

Definiert in Zeile 20 der Datei ModVis2MenuContributor.java.

String getTargetSubmenu ()

Definiert in Zeile 24 der Datei ModVis2MenuContributor.java.

10.2 Schnittstelle zur Analyse

10.2.1 AnalysisAdapter Klassenreferenz

Abgeleitet von `de::offis::pg::eam::mod_vis_2::adapter::IAnalysisAdapter`.

10.2.1.1 Ausführliche Beschreibung

Autor:

philipp

Definiert in Zeile 20 der Datei `AnalysisAdapter.java`.

Öffentliche Methoden

- `AnalysisAdapter (HashMap< String, String > parameter, ArrayList< DataObject > obj)`
- `IDataCollector getCollector ()`
- `IXmlConverter getConverter ()`
- `AnalysisFactory getFactory ()`

Private Attribute

- `IDataCollector collector`
- `IXmlConverter converter`
- `AnalysisFactory factory`

10.2.1.2 Beschreibung der Konstruktoren und Destruktoren

`AnalysisAdapter (HashMap< String, String > parameter, ArrayList< DataObject > obj)`

Definiert in Zeile 25 der Datei `AnalysisAdapter.java`.

Benutzt `AnalysisAdapter.collector`, `AnalysisAdapter.converter` und `AnalysisAdapter.factory`.

10.2.1.3 Dokumentation der Elementfunktionen

`IDataCollector getCollector ()`

Implementiert `IAnalysisAdapter`.

Definiert in Zeile 32 der Datei `AnalysisAdapter.java`.

Benutzt `AnalysisAdapter.collector`.

Wird benutzt von `TableVisualization.generateMatrixTable()` und `TableVisualization.generateTreeTable()`.

IXmlConverter getConverter ()

Implementiert `IAnalysisAdapter`.

Definiert in Zeile 38 der Datei `AnalysisAdapter.java`.

Benutzt `AnalysisAdapter.converter`.

AnalysisFactory getFactory ()

Implementiert `IAnalysisAdapter`.

Definiert in Zeile 43 der Datei `AnalysisAdapter.java`.

Benutzt `AnalysisAdapter.factory`.

10.2.1.4 Dokumentation der Datenelemente

IDataCollector collector [private]

Definiert in Zeile 22 der Datei `AnalysisAdapter.java`.

Wird benutzt von `AnalysisAdapter.AnalysisAdapter()` und `AnalysisAdapter.getCollector()`.

IXmlConverter converter [private]

Definiert in Zeile 23 der Datei `AnalysisAdapter.java`.

Wird benutzt von `AnalysisAdapter.AnalysisAdapter()` und `AnalysisAdapter.getConverter()`.

AnalysisFactory factory [private]

Definiert in Zeile 21 der Datei `AnalysisAdapter.java`.

Wird benutzt von `AnalysisAdapter.AnalysisAdapter()` und `AnalysisAdapter.getFactory()`.

10.2.2 IAnalysisAdapter Schnittstellenreferenz

Basisklasse für `AnalysisAdapter`.

10.2.2.1 Ausführliche Beschreibung

Autor:

philipp

Definiert in Zeile 11 der Datei IAnalysisAdapter.java.

Öffentliche Methoden

- IDataCollector getCollector ()
- IXmlConverter getConverter ()
- AnalysisFactory getFactory ()

10.2.2.2 Dokumentation der Elementfunktionen

IDataCollector getCollector ()

Implementiert in AnalysisAdapter.

IXmlConverter getConverter ()

Implementiert in AnalysisAdapter.

AnalysisFactory getFactory ()

Implementiert in AnalysisAdapter.

10.3 Hilfsklassen

10.3.1 Constants Klassenreferenz

10.3.1.1 Ausführliche Beschreibung

Autor:

philipp In dieser Klasse werden Konstanten definiert und gesammelt, die für das Visualisierungsmodul ModVis2 benutzt werden.

Definiert in Zeile 11 der Datei Constants.java.

Statische öffentliche Attribute

- static final String CONFIG_FILE_NAME = "config_eam_mod_vis.xml"
- static final String MODULE_NAME = "ModVis2"
- static final String PATH_TO_CONFIG_FILE =
System.getProperty("user.dir")+SEP+"eamconfig"+SEP+CONFIG_FILE_NAME
- static final String PATH_TO_INCLUDES = "includes"
- static final String PATH_TO_POPUPS = "popup"
- static final String SEP = System.getProperty("file.separator")

10.3.1.2 Dokumentation der Datenelemente

final String CONFIG_FILE_NAME = "config_eam_mod_vis.xml" [static]

Gibt den Namen der Datei an in der der Pfad zu ModVis1 und ModVis2 hinterlegt ist
Definiert in Zeile 20 der Datei Constants.java.

final String MODULE_NAME = "ModVis2" [static]

Modulname im Webkontext, also der Name des Ordners Webroot...
Definiert in Zeile 14 der Datei Constants.java.

**final String PATH_TO_CONFIG_FILE =
System.getProperty("user.dir")+SEP+"eamconfig"+SEP+CONFIG_FILE_NAME
[static]**

Gibt den Pfad (inkl. Dateiname) an unter der die Konfigurationsdatei mit den Pfaden zu
ModVis1 / ModVis2 zu finden ist.

Definiert in Zeile 24 der Datei Constants.java.

final String PATH_TO_INCLUDES = "includes" [static]

Hier werden die generierten Tabellen gelagert
Definiert in Zeile 31 der Datei Constants.java.

final String PATH_TO_POPUPS = "popup" [static]

Gibt den Pfad zu dem Ordner an in dem die generierten Pop-Up-Seiten befinden
Definiert in Zeile 28 der Datei Constants.java.

final String SEP = System.getProperty("file.separator") [static]

Erzeugt das Zeichen zum Trennen von Ordnern "/" (Linux) oder "\" (Windows)
Definiert in Zeile 17 der Datei Constants.java.

10.3.2 Util Klassenreferenz

10.3.2.1 Ausführliche Beschreibung

In dieser Klasse werden Methoden gesammelt, die das Leben in ModVis2 vereinfachen.

Autor:

philipp

Definiert in Zeile 14 der Datei Util.java.

Öffentliche, statische Methoden

- static String getPathToModVis2 ()

10.3.2.2 Dokumentation der Elementfunktionen

static String getPathToModVis2 () [static]

Diese Methode liefert den Pfad zum Verzeichnis zu de.offis.pg.eam.mod_vis_2 auf Grundlage dessen, was in der im Verzeichnis System.getProperty("user.dir") liegenden "config_eam_mod_vis.xml" steht.

Rückgabe:

Verzeichnispfad

Definiert in Zeile 24 der Datei Util.java.

10.4 Implementierung**10.4.1 Generator Klassenreferenz****10.4.1.1 Ausführliche Beschreibung**

Die Methoden in dieser Klasse generieren die Tabellen von ModVis2

Autor:

philipp

Definiert in Zeile 18 der Datei Generator.java.

Öffentliche, statische Methoden

- static void matrixTable (IDataCollector treeCollector, File file)
- static void treeTable (TreeObject node, File file)

Private, statische Methoden

- static void recursiveTable (TreeObject node, FileWriter out, int numberOfChildren, long level)

Statische private Attribute

- static final String linkEditInstanceObject =
"../extendDataInput/show_ObjectInstance.jsf?"
- static final String linkMetaObjectId = "&metaobjectId="

- `static final String linkObjectId = "&instanceOfMetaobject="`

10.4.1.2 Dokumentation der Elementfunktionen

static void matrixTable (IDataCollector treeCollector, File file) [static]

Diese Methode generiert die HTML-Tabelle in die angegebene Datei.

Parameter:

node Zu diesem Baum soll die Matrix-Tabelle erstellt werden.

file In diese Datei wird die Tabelle geschrieben.

Definiert in Zeile 33 der Datei Generator.java.

Benutzt Generator.linkEditInstanceObject, Generator.linkMetaObjectId und Generator.linkObjectId.

static void recursiveTable (TreeObject node, FileWriter out, int numberOfChildren, long level) [static, private]

Diese Methode generiert das innere der Verbindungsbaum-Tabelle

Siehe auch:

Generator.generateTreeTable

Parameter:

node Das ist der Vaterknoten mit dem alles beginnt.

out In diese Datei soll alles geschrieben werden.

numberOfChildren Gibt an wieviele Kinder ein Knoten hat. Das ist wichtig zu wissen, damit man bei colspan den Richtigen Wert angeben kann, um den Tabellenbaum schön aussehen zu lassen.

Definiert in Zeile 167 der Datei Generator.java.

Benutzt Generator.linkEditInstanceObject, Generator.linkMetaObjectId und Generator.linkObjectId.

Wird benutzt von Generator.treeTable().

static void treeTable (TreeObject node, File file) [static]

Diese Methode schreibt die HTML-Tabelle in die angegebene Datei.

Parameter:

node Zu diesem Knoten soll die Baum-Tabelle erstellt werden.

file In diese Datei wird die Tabelle geschrieben.

Definiert in Zeile 115 der Datei Generator.java.

Benutzt `Generator.linkEditInstanceObject`, `Generator.linkMetaObjectId`, `Generator.linkObjectId` und `Generator.recursiveTable()`.

10.4.1.3 Dokumentation der Datenelemente

```
final String linkEditInstanceObject =  
"../extendDataInput/show_ObjectInstance.jsf?" [static, private]
```

Definiert in Zeile 25 der Datei `Generator.java`.

Wird benutzt von `Generator.matrixTable()`, `Generator.recursiveTable()` und `Generator.treeTable()`.

```
final String linkMetaObjectId = "&metaobjectId=" [static, private]
```

Definiert in Zeile 23 der Datei `Generator.java`.

Wird benutzt von `Generator.matrixTable()`, `Generator.recursiveTable()` und `Generator.treeTable()`.

```
final String linkObjectId = "&instanceOfMetaobject=" [static, private]
```

Definiert in Zeile 24 der Datei `Generator.java`.

Wird benutzt von `Generator.matrixTable()`, `Generator.recursiveTable()` und `Generator.treeTable()`.

10.4.2 Relation Klassenreferenz

10.4.2.1 Ausführliche Beschreibung

Diese Klasse stellt Methoden zur Verfügung, mit denen die Matrix-Relation (Verbindungsmatrix) für die Visualisierung aus Bäumen der Analyse erzeugt werden kann.

Autor:

philipp

Definiert in Zeile 16 der Datei `Relation.java`.

Öffentliche, statische Methoden

- `static String[][] computeMatrixFromTree (List< TreeObject > objects)`
- `static List< TreeObject > getObjectsFromTree (TreeObject tree)`

10.4.2.2 Dokumentation der Elementfunktionen

static String [][] computeMatrixFromTree (List< TreeObject > objects) [static]

Erzeugt die Verbindungsmatrix zwischen den übergebenen TreeObjects

Parameter:

objects Die Liste der Objekte zu denen die Matrix erzeugt werden soll

Rückgabe:

Die Matrix

Definiert in Zeile 73 der Datei Relation.java.

static List<TreeObject> getObjectsFromTree (TreeObject tree) [static]

Liefert alle TreeObjects aus einem Baum

Parameter:

tree Der Baum, der übergeben wird.

Rückgabe:

Liste von TreeObjects

Definiert in Zeile 27 der Datei Relation.java.

10.4.3 SelectObject Klassenreferenz

10.4.3.1 Ausführliche Beschreibung

Autor:

philipp

Definiert in Zeile 38 der Datei SelectObject.java.

Öffentliche Methoden

- String analyse ()
- String getCostTypeObject ()
- HashMap< String, String > getMap ()
- String getPerson ()
- String getRelationPerson ()
- String getSelectedDisplaystyle ()
- String getSelectedInstance ()
- String getSelectedMetamodel ()
- String getSelectedObject ()

- `LinkedList< SelectItem > getSelectOneOfTheseDisplaystyles ()`
- `LinkedList< SelectItem > getSelectOneOfTheseInstances ()`
- `LinkedList< SelectItem > getSelectOneOfTheseMetamodels ()`
- `LinkedList< SelectItem > getSelectOneOfTheseObjects ()`
- `boolean isMatrix ()`
- `boolean isTree ()`
- `void selectDisplaystyle (ValueChangeEvent valueChangeEvent)`
- `void selectInstance (ValueChangeEvent valueChangeEvent)`
- `void selectMetamodel (ValueChangeEvent valueChangeEvent)`
- `void selectObject (ValueChangeEvent valueChangeEvent)`
- `SelectObject ()`
- `void setCostTypeObject (String costTypeObject)`
- `void setMap (HashMap< String, String > map)`
- `void setMatrix (boolean matrix)`
- `void setPerson (String person)`
- `void setRelationPerson (String relationPerson)`
- `void setSelectedDisplaystyle (String selectedDisplaystyle)`
- `void setSelectedInstance (String instance)`
- `void setSelectedMetamodel (String metamodel)`
- `void setSelectedObject (String object)`
- `void setTree (boolean tree)`

Private Methoden

- `void initInstances ()`
- `void initMetamodels ()`
- `void initObjects ()`

Private Attribute

- `final EAMCategoryDAO categoryDAO = DAOFactory.getAuthInstance(userId, password).getEAMCategoryDAO()`
- `String costTypeObject = "server"`
- `final EAMObjectDAO eamObjectDAO = DAOFactory.getAuthInstance(userId, password).getEAMObjectDAO()`
- `final EAMRelationDAO eamRelationDAO = DAOFactory.getAuthInstance(userId, password).getEAMRelationDAO()`
- `DataObject instance = null`
- `HashMap< String, String > map = new HashMap<String, String>()`

- boolean matrix = false
- EAMCategory metamodel = null
- EAMSuperClass object = null
- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String person = "verantwortlicher"
- String relationPerson = "hat"
- String selectedDisplaystyle = null
- String selectedInstance = null
- String selectedMetamodel = null
- String selectedObject = null
- LinkedList< SelectItem > selectOneOfTheseDisplaystyles = new
LinkedList<SelectItem>()
- LinkedList< SelectItem > selectOneOfTheseInstances = new
LinkedList<SelectItem>()
- LinkedList< SelectItem > selectOneOfTheseMetamodels = new
LinkedList<SelectItem>()
- LinkedList< SelectItem > selectOneOfTheseObjects = new
LinkedList<SelectItem>()
- boolean tree = false
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

10.4.3.2 Beschreibung der Konstruktoren und Destruktoren

SelectObject ()

Definiert in Zeile 79 der Datei SelectObject.java.

Benutzt SelectObject.selectOneOfTheseDisplaystyles.

10.4.3.3 Dokumentation der Elementfunktionen

String analyse ()

Definiert in Zeile 385 der Datei SelectObject.java.

Benutzt SelectObject.costTypeObject, SelectObject.instance, SelectObject.isMatrix(),
SelectObject.isTree(), SelectObject.map, SelectObject.person und
SelectObject.relationPerson.

String getCostTypeObject ()

Definiert in Zeile 520 der Datei SelectObject.java.

Benutzt `SelectObject.costTypeObject`.

HashMap<String, String> getMap ()

Definiert in Zeile 502 der Datei `SelectObject.java`.

Benutzt `SelectObject.map`.

String getPerson ()

Definiert in Zeile 508 der Datei `SelectObject.java`.

Benutzt `SelectObject.person`.

String getRelationPerson ()

Definiert in Zeile 514 der Datei `SelectObject.java`.

Benutzt `SelectObject.relationPerson`.

String getSelectedDisplaystyle ()

Definiert in Zeile 494 der Datei `SelectObject.java`.

Benutzt `SelectObject.selectedDisplaystyle`.

String getSelectedInstance ()

Definiert in Zeile 232 der Datei `SelectObject.java`.

Benutzt `SelectObject.selectedInstance`.

String getSelectedMetamodel ()

Definiert in Zeile 89 der Datei `SelectObject.java`.

Benutzt `SelectObject.selectedMetamodel`.

String getSelectedObject ()

Definiert in Zeile 144 der Datei `SelectObject.java`.

Benutzt `SelectObject.selectedObject`.

LinkedList<SelectItem> getSelectOneOfTheseDisplaystyles ()

Definiert in Zeile 347 der Datei `SelectObject.java`.

Benutzt `SelectObject.selectOneOfTheseDisplaystyles`.

LinkedList<SelectItem> getSelectOneOfTheseInstances ()

Definiert in Zeile 326 der Datei SelectObject.java.

Benutzt SelectObject.selectOneOfTheseInstances.

LinkedList<SelectItem> getSelectOneOfTheseMetamodels ()

Definiert in Zeile 116 der Datei SelectObject.java.

Benutzt SelectObject.initMetamodels() und SelectObject.selectOneOfTheseMetamodels.

LinkedList<SelectItem> getSelectOneOfTheseObjects ()

Definiert in Zeile 212 der Datei SelectObject.java.

Benutzt SelectObject.selectOneOfTheseObjects.

void initInstances () [private]

Definiert in Zeile 242 der Datei SelectObject.java.

Benutzt SelectObject.password, SelectObject.selectOneOfTheseInstances und SelectObject.userId.

Wird benutzt von SelectObject.setSelectedObject().

void initMetamodels () [private]

Definiert in Zeile 125 der Datei SelectObject.java.

Benutzt SelectObject.categoryDAO und SelectObject.selectOneOfTheseMetamodels.

Wird benutzt von SelectObject.getSelectOneOfTheseMetamodels().

void initObjects () [private]

Definiert in Zeile 158 der Datei SelectObject.java.

Benutzt SelectObject.eamObjectDAO, SelectObject.metamodel und SelectObject.selectOneOfTheseObjects.

Wird benutzt von SelectObject.setSelectedMetamodel().

boolean isMatrix ()

Definiert in Zeile 365 der Datei SelectObject.java.

Benutzt SelectObject.matrix.

Wird benutzt von SelectObject.analyse().

boolean isTree ()

Definiert in Zeile 371 der Datei SelectObject.java.

Benutzt SelectObject.tree.

Wird benutzt von SelectObject.analyse().

void selectDisplaystyle (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 333 der Datei SelectObject.java.

Benutzt SelectObject.selectedDisplaystyle, SelectObject.setMatrix(), SelectObject.setSelectedDisplaystyle() und SelectObject.setTree().

void selectInstance (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 278 der Datei SelectObject.java.

Benutzt SelectObject.instance, SelectObject.password, SelectObject.selectedInstance, SelectObject.setSelectedInstance() und SelectObject.userId.

void selectMetamodel (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 108 der Datei SelectObject.java.

Benutzt SelectObject.categoryDAO, SelectObject.metamodel, SelectObject.selectedMetamodel und SelectObject.setSelectedMetamodel().

void selectObject (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 188 der Datei SelectObject.java.

Benutzt SelectObject.eamObjectDAO, SelectObject.eamRelationDAO, SelectObject.selectedObject und SelectObject.setSelectedObject().

void setCostTypeObject (String costTypeObject)

Definiert in Zeile 523 der Datei SelectObject.java.

void setMap (HashMap< String, String > map)

Definiert in Zeile 505 der Datei SelectObject.java.

void setMatrix (boolean matrix)

Definiert in Zeile 368 der Datei SelectObject.java.

Wird benutzt von SelectObject.selectDisplaystyle().

void setPerson (String person)

Definiert in Zeile 511 der Datei SelectObject.java.

void setRelationPerson (String relationPerson)

Definiert in Zeile 517 der Datei SelectObject.java.

void setSelectedDisplaystyle (String selectedDisplaystyle)

Definiert in Zeile 497 der Datei SelectObject.java.

Wird benutzt von SelectObject.selectDisplaystyle().

void setSelectedInstance (String instance)

Definiert in Zeile 236 der Datei SelectObject.java.

Benutzt SelectObject.selectedInstance.

Wird benutzt von SelectObject.selectInstance().

void setSelectedMetamodel (String metamodel)

Definiert in Zeile 92 der Datei SelectObject.java.

Benutzt SelectObject.initObjects(), SelectObject.instance, SelectObject.selectedInstance, SelectObject.selectedMetamodel, SelectObject.selectedObject, SelectObject.selectOneOfTheseInstances und SelectObject.selectOneOfTheseObjects.

Wird benutzt von SelectObject.selectMetamodel().

void setSelectedObject (String object)

Definiert in Zeile 148 der Datei SelectObject.java.

Benutzt SelectObject.initInstances() und SelectObject.selectedObject.

Wird benutzt von SelectObject.selectObject().

void setTree (boolean tree)

Definiert in Zeile 374 der Datei SelectObject.java.

Wird benutzt von SelectObject.selectDisplaystyle().

10.4.3.4 Dokumentation der Datenelemente

final EAMCategoryDAO categoryDAO = DAOFactory.getAuthInstance(userId, password).getEAMCategoryDAO() [private]

Definiert in Zeile 45 der Datei SelectObject.java.

Wird benutzt von SelectObject.initMetamodels() und SelectObject.selectMetamodel().

String costTypeObject = "server" [private]

Definiert in Zeile 74 der Datei SelectObject.java.

Wird benutzt von SelectObject.analyse() und SelectObject.getCostTypeObject().

final EAMObjectDAO eamObjectDAO = DAOFactory.getAuthInstance(userId, password).getEAMObjectDAO() [private]

Definiert in Zeile 46 der Datei SelectObject.java.

Wird benutzt von SelectObject.initObjects() und SelectObject.selectObject().

final EAMRelationDAO eamRelationDAO = DAOFactory.getAuthInstance(userId, password).getEAMRelationDAO() [private]

Definiert in Zeile 47 der Datei SelectObject.java.

Wird benutzt von SelectObject.selectObject().

DataObject instance = null [private]

Definiert in Zeile 58 der Datei SelectObject.java.

Wird benutzt von SelectObject.analyse(), SelectObject.selectInstance() und SelectObject.setSelectedMetamodel().

HashMap<String, String> map = new HashMap<String, String>() [private]

Definiert in Zeile 71 der Datei SelectObject.java.

Wird benutzt von SelectObject.analyse() und SelectObject.getMap().

boolean matrix = false [private]

Definiert in Zeile 67 der Datei SelectObject.java.

Wird benutzt von SelectObject.isMatrix().

EAMCategory metamodel = null [private]

Definiert in Zeile 50 der Datei SelectObject.java.

Wird benutzt von SelectObject.initObjects() und SelectObject.selectMetamodel().

EAMSuperClass object = null [private]

Definiert in Zeile 54 der Datei SelectObject.java.

String password =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
[private]

Definiert in Zeile 42 der Datei SelectObject.java.

Wird benutzt von SelectObject.initInstances() und SelectObject.selectInstance().

String person = "verantwortlicher" [private]

Definiert in Zeile 72 der Datei SelectObject.java.

Wird benutzt von SelectObject.analyse() und SelectObject.getPerson().

String relationPerson = "hat" [private]

Definiert in Zeile 73 der Datei SelectObject.java.

Wird benutzt von SelectObject.analyse() und SelectObject.getRelationPerson().

String selectedDisplaystyle = null [private]

Definiert in Zeile 62 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedDisplaystyle() und SelectObject.selectDisplaystyle().

String selectedInstance = null [private]

Definiert in Zeile 59 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedInstance(), SelectObject.selectInstance(), SelectObject.setSelectedInstance() und SelectObject.setSelectedMetamodel().

String selectedMetamodel = null [private]

Definiert in Zeile 51 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedMetamodel(), SelectObject.selectMetamodel() und SelectObject.setSelectedMetamodel().

String selectedObject = null [private]

Definiert in Zeile 55 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedObject(), SelectObject.selectObject(), SelectObject.setSelectedMetamodel() und SelectObject.setSelectedObject().


```
LinkedList<SelectItem> selectOneOfTheseDisplaystyles = new  
LinkedList<SelectItem>() [private]
```

Definiert in Zeile 63 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedOneOfTheseDisplaystyles() und SelectObject.SelectObject().

```
LinkedList<SelectItem> selectOneOfTheseInstances = new  
LinkedList<SelectItem>() [private]
```

Definiert in Zeile 60 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedOneOfTheseInstances(), SelectObject.initInstances() und SelectObject.setSelectedMetamodel().

```
LinkedList<SelectItem> selectOneOfTheseMetamodels = new  
LinkedList<SelectItem>() [private]
```

Definiert in Zeile 52 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedOneOfTheseMetamodels() und SelectObject.initMetamodels().

```
LinkedList<SelectItem> selectOneOfTheseObjects = new  
LinkedList<SelectItem>() [private]
```

Definiert in Zeile 56 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedOneOfTheseObjects(), SelectObject.initObjects() und SelectObject.setSelectedMetamodel().

```
boolean tree = false [private]
```

Definiert in Zeile 68 der Datei SelectObject.java.

Wird benutzt von SelectObject.isTree().

```
String userId =  
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")  
[private]
```

Definiert in Zeile 41 der Datei SelectObject.java.

Wird benutzt von SelectObject.initInstances() und SelectObject.selectInstance().

10.4.4 TableVisualization Klassenreferenz

10.4.4.1 Ausführliche Beschreibung

Diese Klasse stellt die notwendigen Methoden zur Verfügung, die benötigt werden, um die Inhalte auf den JSPs von ModVis2 darstellen zu können

Autor:

philipp

Definiert in Zeile 22 der Datei TableVisualization.java.

Öffentliche Methoden

- String getHeadLineName ()
- DataObject getInstance ()
- String getMatrixFilename ()
- String getTreeFilename ()
- boolean isMatrix ()
- boolean isTree ()

Öffentliche, statische Methoden

- static void generateMatrixTable ()
- static void generateTreeTable ()
- static HashMap< String, String > getMap ()
- static void setHeadLineName (String name)
- static void setInstance (DataObject instance)
- static void setMap (HashMap< String, String > map)
- static void setMatrix (boolean m)
- static void setTree (boolean t)

Statische private Attribute

- static String headLineName = ""
- static DataObject instance = null
- static HashMap< String, String > map = null
- static boolean matrix = false
- static String sep = Constants.SEP
- static boolean tree = false

10.4.4.2 Dokumentation der Elementfunktionen

static void generateMatrixTable () [static]

Diese Methode generiert eine HTML-Tabelle. Die Tabelle stellt die Verbindungsstrukturen zwischen den einzelnen Objekten eines Baumes dar.

Rückgabe:

mod2SuccessMatrix Success-String für die Navigation

Definiert in Zeile 45 der Datei TableVisualization.java.

Benutzt AnalysisAdapter.getCollector(), TableVisualization.instance, TableVisualization.map und TableVisualization.sep.

static void generateTreeTable () [static]

Diese Methode generiert eine HTML-Tabelle, die den Verbindungsbaum darstellt.

Rückgabe:

mod2SuccessTree Success-String für die Navigation in der faces-config.xml

Definiert in Zeile 84 der Datei TableVisualization.java.

Benutzt AnalysisAdapter.getCollector(), TableVisualization.instance und TableVisualization.sep.

String getHeadLineName ()

Definiert in Zeile 131 der Datei TableVisualization.java.

Benutzt TableVisualization.headLineName.

DataObject getInstance ()

Definiert in Zeile 158 der Datei TableVisualization.java.

Benutzt TableVisualization.instance.

static HashMap<String, String> getMap () [static]

Definiert in Zeile 137 der Datei TableVisualization.java.

Benutzt TableVisualization.map.

String getMatrixFilename ()

Definiert in Zeile 147 der Datei TableVisualization.java.

String getTreeFilename ()

Definiert in Zeile 152 der Datei TableVisualization.java.

boolean isMatrix ()

Definiert in Zeile 168 der Datei TableVisualization.java.

Benutzt TableVisualization.matrix.

boolean isTree ()

Definiert in Zeile 176 der Datei TableVisualization.java.

Benutzt TableVisualization.tree.

static void setHeadLineName (String name) [static]

Definiert in Zeile 127 der Datei TableVisualization.java.

Benutzt TableVisualization.headLineName.

static void setInstance (DataObject instance) [static]

Definiert in Zeile 163 der Datei TableVisualization.java.

static void setMap (HashMap< String, String > map) [static]

Definiert in Zeile 141 der Datei TableVisualization.java.

static void setMatrix (boolean m) [static]

Definiert in Zeile 172 der Datei TableVisualization.java.

Benutzt TableVisualization.matrix.

static void setTree (boolean t) [static]

Definiert in Zeile 180 der Datei TableVisualization.java.

Benutzt TableVisualization.tree.

10.4.4.3 Dokumentation der Datenelemente

String headLineName = "" [static, private]

Definiert in Zeile 33 der Datei TableVisualization.java.

Wird benutzt von TableVisualization.getHeadLineName() und TableVisualization.setHeadLineName().

DBObject instance = null [static, private]

Definiert in Zeile 27 der Datei TableVisualization.java.

Wird benutzt von TableVisualization.generateMatrixTable(),
TableVisualization.generateTreeTable() und TableVisualization.getInstance().

HashMap<String, String> map = null [static, private]

Definiert in Zeile 31 der Datei TableVisualization.java.

Wird benutzt von TableVisualization.generateMatrixTable() und
TableVisualization.getMap().

boolean matrix = false [static, private]

Definiert in Zeile 28 der Datei TableVisualization.java.

Wird benutzt von TableVisualization.isMatrix() und TableVisualization.setMatrix().

String sep = Constants.SEP [static, private]

Definiert in Zeile 24 der Datei TableVisualization.java.

Wird benutzt von TableVisualization.generateMatrixTable() und
TableVisualization.generateTreeTable().

boolean tree = false [static, private]

Definiert in Zeile 29 der Datei TableVisualization.java.

Wird benutzt von TableVisualization.isTree() und TableVisualization.setTree().

11 de.offis.pg.eam.mod_vis_3

11.1 BundleConfiguration

11.1.1 BundleViews Klassenreferenz

Abgeleitet von `de::offis::pg::eam::core::module::api::IModuleView`.

11.1.1.1 Ausführliche Beschreibung

Definiert in Zeile 21 der Datei `BundleViews.java`.

Öffentliche Methoden

- `String getBundleVersion ()`
- `String getDescription ()`
- `long getEquinoxBundleId ()`
- `String getName ()`
- `List< View > getViews ()`

Statische private Attribute

- `static final String VIEWS = "bundle_views.xml"`

11.1.1.2 Dokumentation der Elementfunktionen

String getBundleVersion ()

Definiert in Zeile 67 der Datei `BundleViews.java`.

String getDescription ()

Definiert in Zeile 27 der Datei `BundleViews.java`.

long getEquinoxBundleId ()

Definiert in Zeile 32 der Datei `BundleViews.java`.

String getName ()

Definiert in Zeile 37 der Datei BundleViews.java.

List<View> getViews ()

Definiert in Zeile 42 der Datei BundleViews.java.

Benutzt BundleViews.VIEWS.

11.1.1.3 Dokumentation der Datenelemente

final String VIEWS = "bundle_views.xml" [static, private]

Definiert in Zeile 23 der Datei BundleViews.java.

Wird benutzt von BundleViews.getViews().

11.1.2 Configuration Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::configuration::api::IBundleConfiguration.

11.1.2.1 Ausführliche Beschreibung

Definiert in Zeile 10 der Datei Configuration.java.

Öffentliche Methoden

- boolean bundleHasObjects ()
- boolean bundleHasRelations ()
- boolean bundleHasViews ()
- Document getObjectsJdomDocument () throws JDOMException, IOException

11.1.2.2 Dokumentation der Elementfunktionen

boolean bundleHasObjects ()

Definiert in Zeile 13 der Datei Configuration.java.

boolean bundleHasRelations ()

Definiert in Zeile 28 der Datei Configuration.java.

boolean bundleHasViews ()

Definiert in Zeile 18 der Datei Configuration.java.

Document `getObjectsJdomDocument ()` throws `JDOMException`, `IOException`

Definiert in Zeile 23 der Datei `Configuration.java`.

11.1.3 Activator Klassenreferenz

Abgeleitet von `org::osgi::framework::BundleActivator`.

11.1.3.1 Ausführliche Beschreibung

Definiert in Zeile 30 der Datei `Activator.java`.

Öffentliche Methoden

- `void start (BundleContext context) throws Exception`
- `void stop (BundleContext context) throws Exception`

Statische öffentliche Attribute

- `static String bundleSymbolicName`
- `static String bundleVersion`
- `static long equinoxBundleId`

Private Attribute

- `ServiceTracker httpServiceTracker`

Klassen

- `class HttpServiceTracker`
- `class ServletContextListenerServletAdaptor`

11.1.3.2 Dokumentation der Elementfunktionen**`void start (BundleContext context) throws Exception`**

Definiert in Zeile 38 der Datei `Activator.java`.

Benutzt `Activator.bundleSymbolicName`, `Activator.bundleVersion`, `Activator.equinoxBundleId` und `Activator.httpServiceTracker`.

void stop (BundleContext context) throws Exception

Definiert in Zeile 47 der Datei Activator.java.

Benutzt Activator.httpServiceTracker.

11.1.3.3 Dokumentation der Datenelemente

String bundleSymbolicName [static]

Definiert in Zeile 34 der Datei Activator.java.

Wird benutzt von Activator.start().

String bundleVersion [static]

Definiert in Zeile 35 der Datei Activator.java.

Wird benutzt von Activator.start().

long equinoxBundleId [static]

Definiert in Zeile 36 der Datei Activator.java.

Wird benutzt von Activator.start().

ServiceTracker httpServiceTracker [private]

Definiert in Zeile 32 der Datei Activator.java.

Wird benutzt von Activator.start() und Activator.stop().

11.1.4 Activator.HttpServiceTracker Klassenreferenz

Abgeleitet von org::osgi::util::tracker::ServiceTracker.

11.1.4.1 Ausführliche Beschreibung

Definiert in Zeile 51 der Datei Activator.java.

Öffentliche Methoden

- Object addingService (ServiceReference reference)
- HttpServiceTracker (BundleContext context)
- void removedService (ServiceReference reference, Object service)

Statische private Attribute

- static final String PATH = "/" + Constants.MODULE_NAME
- static final String WEB_ROOT = "/WebRoot"

11.1.4.2 Beschreibung der Konstruktoren und Destruktoren

HttpServiceTracker (BundleContext context)

Definiert in Zeile 56 der Datei Activator.java.

11.1.4.3 Dokumentation der Elementfunktionen

Object addingService (ServiceReference reference)

Definiert in Zeile 60 der Datei Activator.java.

Benutzt Activator.HttpServiceTracker.PATH und Activator.HttpServiceTracker.WEB_ROOT.

void removedService (ServiceReference reference, Object service)

Definiert in Zeile 95 der Datei Activator.java.

Benutzt Activator.HttpServiceTracker.PATH.

11.1.4.4 Dokumentation der Datenelemente

final String PATH = "/" + Constants.MODULE_NAME [static, private]

Definiert in Zeile 53 der Datei Activator.java.

Wird benutzt von Activator.HttpServiceTracker.addingService() und Activator.HttpServiceTracker.removedService().

final String WEB_ROOT = "/WebRoot" [static, private]

Definiert in Zeile 54 der Datei Activator.java.

Wird benutzt von Activator.HttpServiceTracker.addingService().

11.1.5 Activator.ServletContextListenerServletAdaptor Klassenreferenz

Abgeleitet von javax::servlet::Servlet.

11.1.5.1 Ausführliche Beschreibung

Definiert in Zeile 104 der Datei Activator.java.

Öffentliche Methoden

- void destroy ()
- ServletConfig getServletConfig ()
- String getServletInfo ()
- void init (ServletConfig config) throws ServletException
- void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException
- ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Private Attribute

- ServletConfig config
- Servlet delegate
- ClassLoader jspLoader
- ServletContextListener listener

11.1.5.2 Beschreibung der Konstruktoren und Destruktoren

ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Definiert in Zeile 110 der Datei Activator.java.

11.1.5.3 Dokumentation der Elementfunktionen

void destroy ()

Definiert in Zeile 144 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config, Activator.ServletContextListenerServletAdaptor.delegate, Activator.ServletContextListenerServletAdaptor.jspLoader und Activator.ServletContextListenerServletAdaptor.listener.

ServletConfig getServletConfig ()

Definiert in Zeile 158 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config.

String getServletInfo ()

Definiert in Zeile 162 der Datei Activator.java.

void init (ServletConfig config) throws ServletException

Definiert in Zeile 118 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config,
Activator.ServletContextListenerServletAdaptor.delegate,
Activator.ServletContextListenerServletAdaptor.jspLoader und
Activator.ServletContextListenerServletAdaptor.listener.

void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException

Definiert in Zeile 132 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.delegate und
Activator.ServletContextListenerServletAdaptor.jspLoader.

11.1.5.4 Dokumentation der Datenelemente**ServletConfig config** [private]

Definiert in Zeile 105 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.getServletConfig() und
Activator.ServletContextListenerServletAdaptor.init().

Servlet delegate [private]

Definiert in Zeile 107 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.init() und
Activator.ServletContextListenerServletAdaptor.service().

ClassLoader jspLoader [private]

Definiert in Zeile 108 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.init() und
Activator.ServletContextListenerServletAdaptor.service().

ServletContextListener listener [private]

Definiert in Zeile 106 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy() und
Activator.ServletContextListenerServletAdaptor.init().

11.1.6 ModVis3MenuContributor Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::api::AbstractAdoptableMenuContributor.

11.1.6.1 Ausführliche Beschreibung

Autor:

philipp

Definiert in Zeile 13 der Datei ModVis3MenuContributor.java.

Öffentliche Methoden

- String getAdopterId ()
- String getContributionTarget ()
- String getContributorId ()
- String getTargetSubmenu ()
- ModVis3MenuContributor ()

11.1.6.2 Beschreibung der Konstruktoren und Destruktoren

ModVis3MenuContributor ()

Definiert in Zeile 15 der Datei ModVis3MenuContributor.java.

11.1.6.3 Dokumentation der Elementfunktionen

String getAdopterId ()

Definiert in Zeile 19 der Datei ModVis3MenuContributor.java.

String getContributionTarget ()

Definiert in Zeile 23 der Datei ModVis3MenuContributor.java.

String getContributorId ()

Definiert in Zeile 27 der Datei ModVis3MenuContributor.java.

String getTargetSubmenu ()

Definiert in Zeile 31 der Datei ModVis3MenuContributor.java.

11.2 Schnittstelle zur Analyse

11.2.1 AnalysisAdapter Klassenreferenz

Abgeleitet von `de::offis::pg::eam::mod_vis_3::adapter::IAnalysisAdapter`.

11.2.1.1 Ausführliche Beschreibung

Autor:

philipp

Definiert in Zeile 21 der Datei `AnalysisAdapter.java`.

Öffentliche Methoden

- `AnalysisAdapter (HashMap< String, String > parameter, ArrayList< DataObject > obj)`
- `IDataCollector getCollector ()`
- `IXmlConverter getConverter ()`
- `AnalysisFactory getFactory ()`

Private Attribute

- `IDataCollector collector`
- `IXmlConverter converter`
- `AnalysisFactory factory`

11.2.1.2 Beschreibung der Konstruktoren und Destruktoren

`AnalysisAdapter (HashMap< String, String > parameter, ArrayList< DataObject > obj)`

Definiert in Zeile 26 der Datei `AnalysisAdapter.java`.

Benutzt `AnalysisAdapter.collector`, `AnalysisAdapter.converter` und `AnalysisAdapter.factory`.

11.2.1.3 Dokumentation der Elementfunktionen

`IDataCollector getCollector ()`

Implementiert `IAnalysisAdapter`.

Definiert in Zeile 33 der Datei `AnalysisAdapter.java`.

Benutzt `AnalysisAdapter.collector`.

IXmlConverter getConverter ()

Implementiert IAnalysisAdapter.

Definiert in Zeile 39 der Datei AnalysisAdapter.java.

Benutzt AnalysisAdapter.converter.

Wird benutzt von MyGraph.drawNewSVG().

AnalysisFactory getFactory ()

Implementiert IAnalysisAdapter.

Definiert in Zeile 44 der Datei AnalysisAdapter.java.

Benutzt AnalysisAdapter.factory.

11.2.1.4 Dokumentation der Datenelemente

IDataCollector collector [private]

Definiert in Zeile 23 der Datei AnalysisAdapter.java.

Wird benutzt von AnalysisAdapter.AnalysisAdapter() und AnalysisAdapter.getCollector().

IXmlConverter converter [private]

Definiert in Zeile 24 der Datei AnalysisAdapter.java.

Wird benutzt von AnalysisAdapter.AnalysisAdapter() und AnalysisAdapter.getConverter().

AnalysisFactory factory [private]

Definiert in Zeile 22 der Datei AnalysisAdapter.java.

Wird benutzt von AnalysisAdapter.AnalysisAdapter() und AnalysisAdapter.getFactory().

11.2.2 IAnalysisAdapter Schnittstellenreferenz

Basisklasse für AnalysisAdapter.

11.2.2.1 Ausführliche Beschreibung

Autor:

philipp

Definiert in Zeile 11 der Datei IAnalysisAdapter.java.

Öffentliche Methoden

- `IDataCollector` `getCollector ()`
- `IXmlConverter` `getConverter ()`
- `AnalysisFactory` `getFactory ()`

11.2.2.2 Dokumentation der Elementfunktionen

IDataCollector `getCollector ()`

Implementiert in `AnalysisAdapter`.

IXmlConverter `getConverter ()`

Implementiert in `AnalysisAdapter`.

AnalysisFactory `getFactory ()`

Implementiert in `AnalysisAdapter`.

11.3 Hilfsklassen

11.3.1 Constants Klassenreferenz

11.3.1.1 Ausführliche Beschreibung

Autor:

philipp In dieser Klasse werden Konstanten definiert und gesammelt, die für das Visualisierungsmodul `ModVis3` benutzt werden.

Definiert in Zeile 12 der Datei `Constants.java`.

Statische öffentliche Attribute

- `static final String CONFIG_FILE_NAME = "config_eam_mod_vis.xml"`
- `static final String MODULE_NAME = "ModVis3"`
- `static final long OLD_FILE = 86400000`
- `static final String PATH_TO_CONFIG_FILE = System.getProperty("user.dir")+SEP+"eamconfig"+SEP+CONFIG_FILE_NAME`
- `static final String PATH_TO_IMAGES = "images"`
- `static final String SEP = System.getProperty("file.separator")`

11.3.1.2 Dokumentation der Datenelemente

final String CONFIG_FILE_NAME = "config_eam_mod_vis.xml" [static]

Gibt den Namen der Datei an in der der Pfad zu ModVis1, 2 und 3 hinterlegt ist
Definiert in Zeile 21 der Datei Constants.java.

final String MODULE_NAME = "ModVis3" [static]

Modulname im Webkontext, also der Name des Ordners Webroot...
Definiert in Zeile 15 der Datei Constants.java.

final long OLD_FILE = 86400000 [static]

Dateien, die älter sind, werden beim Starten des ModVis1 gelöscht wird nur bei von ModVis1 generierten (svg / jpg) dateien gemacht TODO: hier wieder den alten wert von 24h einsetzen!

Definiert in Zeile 34 der Datei Constants.java.

**final String PATH_TO_CONFIG_FILE =
System.getProperty("user.dir")+SEP+"eamconfig"+SEP+CONFIG_FILE_NAME
[static]**

Gibt den Pfad (inkl. Dateiname) an unter der die Konfigurationsdatei mit den Pfaden zu ModVis1 / ModVis2 zu finden ist.

Definiert in Zeile 25 der Datei Constants.java.

final String PATH_TO_IMAGES = "images" [static]

Gibt den Pfad zu dem Ordner an in dem die generierten Bilder (SVG, JPG) gespeichert werden.

Definiert in Zeile 29 der Datei Constants.java.

final String SEP = System.getProperty("file.separator") [static]

Erzeugt das Zeichen zum Trennen von Ordnern "/" (Linux) oder "\" (Windows)

Definiert in Zeile 18 der Datei Constants.java.

11.3.2 ExportSVG Klassenreferenz

11.3.2.1 Ausführliche Beschreibung

Wandelt ein svgBild in ein anders Bildformat um. Das neue Bild findet sich im Gleichen Ordner wie das alte Bild, es sei denn, man gibt den Methoden einen neuen Speicherplatz

mit.

Autor:

philipp

Definiert in Zeile 32 der Datei ExportSVG.java.

Öffentliche, statische Methoden

- static void toJPG (String pathToSVGImage, String pathToJPGImage, String jpgName)
- static void toJPG (String pathToSVGImage, String pathToJPGImage)
- static void toJPG (String pathToSVGImage)
- static void toPNG (String path1, String path2)

Private, statische Methoden

- static String getLink (Element elem)
- static void transcode (String file1, String file2)

Statische private Attribute

- static String sep = Constants.SEP

11.3.2.2 Dokumentation der Elementfunktionen

static String getLink (Element elem) [static, private]

Diese Methode bekommt ein `jdom.Element` übergeben. Sie findet heraus, wo das entsprechende PNG liegt, welches das Element im SVG repräsentiert.

Parameter:

elem Das Element

Rückgabe:

String Der Link zum PNG

Definiert in Zeile 291 der Datei ExportSVG.java.

Benutzt ExportSVG.sep.

Wird benutzt von ExportSVG.toPNG().

static void toJPG (String pathToSVGImage, String pathToJPGImage, String jpgName) [static]

Erzeugt aus einem SVG-Bild ein JPG-Bild an einem benutzerdefinierten Ort

Parameter:

pathToSVGImage Hier liegt das SVG-Bild (Voller Pfad)

pathToJPGImage Hier soll das JPG-Bild liegen

jpgName Den Namen soll das neue JPG haben

Definiert in Zeile 108 der Datei ExportSVG.java.

Benutzt ExportSVG.sep.

static void toJPG (String pathToSVGImage, String pathToJPGImage) [static]

Erzeugt aus einem SVG-Bild ein JPG-Bild an einem benutzerdefinierten Ort

Parameter:

pathToSVGImage Hier liegt das SVG-Bild (Voller Pfad)

pathToJPGImage Hier soll das JPG-Bild liegen (Voller Pfad mit Bild-Namen)

Definiert in Zeile 75 der Datei ExportSVG.java.

static void toJPG (String pathToSVGImage) [static]

Erzeugt aus einem SVG-Bild ein JPG-Bild im gleichen Ordner mit gleichem Namen wie das SVG.

Parameter:

pathToSVGImage Hier liegt das SVG-Bild (Voller Pfad)

Definiert in Zeile 42 der Datei ExportSVG.java.

static void toPNG (String path1, String path2) [static]

Exportiert ein SVG-Bild in ein PNG-Bild. Bei den Pfaden müssen die vollständigen Pfade (inkl Dateinamen und Endung) angegeben werden.

Parameter:

file1 Pfad zum SVG-Bild

file2 Pfad zum PNG-Bild

Definiert in Zeile 177 der Datei ExportSVG.java.

Benutzt ExportSVG.getLink() und ExportSVG.transcode().

static void transcode (String file1, String file2) [static, private]

Wandelt ein SVG-Bild in ein PNG-Bild. Bei den Pfaden müssen die vollständigen Pfade (inkl Dateinamen und Endung) angegeben werden.

Parameter:

file1 Pfad zum SVG-Bild

file2 Pfad zum PNG-Bild

Definiert in Zeile 146 der Datei ExportSVG.java.

Wird benutzt von ExportSVG.toPNG().

11.3.2.3 Dokumentation der Datenelemente

String sep = Constants.SEP [static, private]

Definiert in Zeile 34 der Datei ExportSVG.java.

Wird benutzt von ExportSVG.getLink() und ExportSVG.toJPG().

11.3.3 Util Klassenreferenz

11.3.3.1 Ausführliche Beschreibung

Autor:

philipp

Definiert in Zeile 19 der Datei Util.java.

Öffentliche, statische Methoden

- static void copyFile (String src, String dest)
- static void deleteOldSvgPngFiles (File directory)
- static String getPathToModVis3 ()

Private, statische Methoden

- static void copy (InputStream in, OutputStream out) throws IOException

11.3.3.2 Dokumentation der Elementfunktionen

static void copy (InputStream in, OutputStream out) throws IOException [static, private]

Kopiert den InputStream in den OutputStream

Parameter:

in Quelle des Streams

out Ziel des Streams

Ausnahmebehandlung:

IOException

Definiert in Zeile 74 der Datei Util.java.

Wird benutzt von Util.copyFile().

static void copyFile (String src, String dest) [static]

Kopiert mit Hilfe der Methode {

Siehe auch:

copy} die erste Datei am Ort src an den Ort dest.

Parameter:

src Quelldatei

dest Zieldatei

Definiert in Zeile 90 der Datei Util.java.

Benutzt Util.copy().

static void deleteOldSvgPngFiles (File directory) [static]

löscht alle .svg und .jpg und .jsp Dateien aus dem angegebenen Verzeichnis, die älter sind, als in {

Siehe auch:

Constants.OLD_FILE} angegeben (24h).

Parameter:

directory Aus diesem Verzeichnis sollen alle "alten" Dateien entfernt werden.

Definiert in Zeile 54 der Datei Util.java.

static String getPathToModVis3 () [static]

Diese Methode liefert den Pfad zum Verzeichnis zu de.offis.pg.eam.mod_vis_3 auf Grundlage dessen, was in der im Verzeichnis System.getProperty("user.dir") liegenden "config_eam_mod_vis.xml" steht.

Rückgabe:

Verzeichnispfad

Definiert in Zeile 27 der Datei Util.java.

11.4 Implementierung

11.4.1 MyGraph Klassenreferenz

11.4.1.1 Ausführliche Beschreibung

Autor:

Philipp Gringel
Mart Koehler

Definiert in Zeile 37 der Datei MyGraph.java.

Öffentliche Methoden

- String getCurrentUser ()
- String getGraphFilename ()
- String getGraphLegendFilename ()
- String getHeadLineName ()
- String saveAndStoreInContainer ()
- void setCurrentUser (String currentUser)

Öffentliche, statische Methoden

- static String drawNewSVG ()
- static DataObject getInstance ()
- static HashMap< String, String > getMap ()
- static String getSvgName ()
- static void setHeadLineName (String name)
- static void setInstance (DataObject inst)
- static void setMap (HashMap< String, String > map)
- static void setSvgName (String name)

Private Methoden

- String getPngName ()

Private, statische Methoden

- static void buildGraph (Element root, IGraph g)
- static void createGraph (Document doc)

Private Attribute

- `String currentUser = (String) FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")`

Statische private Attribute

- `static String headLineName = ""`
- `static DataObject instance = null`
- `static HashMap< String, String > map = null`
- `static String sep = Constants.SEP`
- `static String svgName = ""`

11.4.1.2 Dokumentation der Elementfunktionen

static void buildGraph (Element root, IGraph g) [static, private]

Baut den eigentlichen Graph auf, indem das Interface IGraph mit den Daten aus Element gefüllt wird.

Parameter:

root Das Wurzelement des Documents mit der Graphbeschreibung.

g Der Graph der aufgebaut werden soll.

Definiert in Zeile 158 der Datei MyGraph.java.

Benutzt `Node.addNodeAttachment()`, `Edge.setColor()`, `Edge.setDashArray()`, `Node.setK()`, `Edge.setLabel()`, `Node.setLabel()`, `NodeAttachment.setLabelref()`, `Edge.setLinkAddress()`, `Node.setLinkAddress()`, `Edge.setOpacity()`, `NodeAttachment.setOpacity()`, `NodeAttachment.setRx()`, `NodeAttachment.setRy()`, `NodeAttachment.setSizeFactor()`, `Edge.setTooltip()`, `NodeAttachment.setTooltip()`, `Node.setTooltip()`, `NodeAttachment.setUrl()` und `Edge.setWidth()`.

Wird benutzt von `MyGraph.createGraph()`.

static void createGraph (Document doc) [static, private]

Diese Methode baut den Graphen als SVG auf

Parameter:

doc In diesem Document ist die xml-Beschreibung des Graphen

Definiert in Zeile 80 der Datei MyGraph.java.

Benutzt `SVGCreatorIcons.addGraph()`, `MyGraph.buildGraph()`, `MyGraph.getSvgName()`, `IGraph.layout()`, `MyGraph.sep` und `SVGCreatorIcons.writeXmlFile()`.

Wird benutzt von `MyGraph.drawNewSVG()`.

static String drawNewSVG () [static]

Diese Methode generiert das SVG-Tree-Bild

Rückgabe:

”done” String der das Ende des Zeichnens anzeigt.

Definiert in Zeile 60 der Datei MyGraph.java.

Benutzt MyGraph.createGraph(), AnalysisAdapter.getConverter(), MyGraph.instance und MyGraph.map.

String getCurrentUser ()

Definiert in Zeile 365 der Datei MyGraph.java.

Benutzt MyGraph.currentUser.

Wird benutzt von MyGraph.saveAndStoreInContainer().

String getGraphFilename ()

Definiert in Zeile 332 der Datei MyGraph.java.

String getGraphLegendFilename ()

Definiert in Zeile 340 der Datei MyGraph.java.

Benutzt MyGraph.sep.

String getHeadLineName ()

Definiert in Zeile 353 der Datei MyGraph.java.

Benutzt MyGraph.headLineName.

static DataObject getInstance () [static]

Definiert in Zeile 375 der Datei MyGraph.java.

Benutzt MyGraph.instance.

static HashMap<String, String> getMap () [static]

Definiert in Zeile 357 der Datei MyGraph.java.

Benutzt MyGraph.map.

String getPngName () [private]

Definiert in Zeile 384 der Datei MyGraph.java.

Benutzt MyGraph.instance.

Wird benutzt von MyGraph.saveAndStoreInContainer().

static String getSvgName () [static]

Definiert in Zeile 389 der Datei MyGraph.java.

Benutzt MyGraph.svgName.

Wird benutzt von MyGraph.createGraph() und MyGraph.saveAndStoreInContainer().

String saveAndStoreInContainer ()

Diese Methode speichert das generierte SVG als PNG datei und legt es dann für den Benutzer in den Container. Dort kann es vom Export-Modul entnommen werden.

Rückgabe:

 null Es ist keine für die Navigation relevante Rückgabe nötig.

Definiert in Zeile 117 der Datei MyGraph.java.

Benutzt MyGraph.getCurrentUser(), MyGraph.getPngName(), MyGraph.getSvgName(), MyGraph.sep und MyGraph.setCurrentUser().

void setCurrentUser (String currentUser)

Definiert in Zeile 369 der Datei MyGraph.java.

Wird benutzt von MyGraph.saveAndStoreInContainer().

static void setHeadLineName (String name) [static]

Definiert in Zeile 349 der Datei MyGraph.java.

Benutzt MyGraph.headLineName.

static void setInstance (DataObject inst) [static]

Definiert in Zeile 379 der Datei MyGraph.java.

Benutzt MyGraph.instance.

static void setMap (HashMap< String, String > map) [static]

Definiert in Zeile 361 der Datei MyGraph.java.

static void setSvgName (String name) [static]

Definiert in Zeile 392 der Datei MyGraph.java.

Benutzt MyGraph.svgName.

11.4.1.3 Dokumentation der Datenelemente

String currentUser = (String)
FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")
 [private]

Definiert in Zeile 42 der Datei MyGraph.java.

Wird benutzt von MyGraph.getCurrentUser().

String headLineName = "" [static, private]

Definiert in Zeile 50 der Datei MyGraph.java.

Wird benutzt von MyGraph.getHeadLineName() und MyGraph.setHeadLineName().

DataObject instance = null [static, private]

Definiert in Zeile 44 der Datei MyGraph.java.

Wird benutzt von MyGraph.drawNewSVG(), MyGraph.getInstance(),
 MyGraph.getPngName() und MyGraph.setInstance().

HashMap<String, String> map = null [static, private]

Definiert in Zeile 48 der Datei MyGraph.java.

Wird benutzt von MyGraph.drawNewSVG() und MyGraph.getMap().

String sep = Constants.SEP [static, private]

Definiert in Zeile 40 der Datei MyGraph.java.

Wird benutzt von MyGraph.createGraph(), MyGraph.getGraphLegendFilename() und
 MyGraph.saveAndStoreInContainer().

String svgName = "" [static, private]

Definiert in Zeile 46 der Datei MyGraph.java.

Wird benutzt von MyGraph.getSvgName() und MyGraph.setSvgName().

11.4.2 SelectObject Klassenreferenz

11.4.2.1 Ausführliche Beschreibung

Autor:

philipp

Definiert in Zeile 39 der Datei SelectObject.java.

Öffentliche Methoden

- String analyse ()
- String getCostTypeObject ()
- HashMap< String, String > getMap ()
- String getPerson ()
- String getRelationPerson ()
- String getSelectedInstance ()
- String getSelectedMetamodel ()
- String getSelectedObject ()
- LinkedList< SelectItem > getSelectOneOfTheseInstances ()
- LinkedList< SelectItem > getSelectOneOfTheseMetamodels ()
- LinkedList< SelectItem > getSelectOneOfTheseObjects ()
- void selectInstance (ValueChangeEvent valueChangeEvent)
- void selectMetamodel (ValueChangeEvent valueChangeEvent)
- void selectObject (ValueChangeEvent valueChangeEvent)
- void setCostTypeObject (String costTypeObject)
- void setMap (HashMap< String, String > map)
- void setPerson (String person)
- void setRelationPerson (String relationPerson)
- void setSelectedInstance (String instance)
- void setSelectedMetamodel (String metamodel)
- void setSelectedObject (String object)

Private Methoden

- void initInstances ()
- void initMetamodels ()
- void initObjects ()
- String readTemplate (String path)

Private Attribute

- final EAMCategoryDAO categoryDAO = DAOFactory.getAuthInstance(userId, password).getEAMCategoryDAO()
- String costTypeObject = "server"
- final EAMObjectDAO eamObjectDAO = DAOFactory.getAuthInstance(userId, password).getEAMObjectDAO()
- final EAMRelationDAO eamRelationDAO = DAOFactory.getAuthInstance(userId, password).getEAMRelationDAO()

- DataObject instance = null
- HashMap< String, String > map = new HashMap<String, String>()
- EAMCategory metamodel = null
- EAMSuperClass object = null
- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String person = "verantwortlicher"
- String relationPerson = "hat"
- String selectedInstance = null
- String selectedMetamodel = null
- String selectedObject = null
- LinkedList< SelectItem > selectOneOfTheseInstances = new
LinkedList<SelectItem>()
- LinkedList< SelectItem > selectOneOfTheseMetamodels = new
LinkedList<SelectItem>()
- LinkedList< SelectItem > selectOneOfTheseObjects = new
LinkedList<SelectItem>()
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

11.4.2.2 Dokumentation der Elementfunktionen

String analyse ()

Definiert in Zeile 311 der Datei SelectObject.java.

Benutzt SelectObject.costTypeObject, SelectObject.instance, SelectObject.map, SelectObject.person, SelectObject.readTemplate() und SelectObject.relationPerson.

String getCostTypeObject ()

Definiert in Zeile 428 der Datei SelectObject.java.

Benutzt SelectObject.costTypeObject.

HashMap<String, String> getMap ()

Definiert in Zeile 410 der Datei SelectObject.java.

Benutzt SelectObject.map.

String getPerson ()

Definiert in Zeile 416 der Datei SelectObject.java.

Benutzt SelectObject.person.

String getRelationPerson ()

Definiert in Zeile 422 der Datei SelectObject.java.

Benutzt SelectObject.relationPerson.

String getSelectedInstance ()

Definiert in Zeile 215 der Datei SelectObject.java.

Benutzt SelectObject.selectedInstance.

String getSelectedMetamodel ()

Definiert in Zeile 72 der Datei SelectObject.java.

Benutzt SelectObject.selectedMetamodel.

String getSelectedObject ()

Definiert in Zeile 127 der Datei SelectObject.java.

Benutzt SelectObject.selectedObject.

LinkedList<SelectItem> getSelectOneOfTheseInstances ()

Definiert in Zeile 303 der Datei SelectObject.java.

Benutzt SelectObject.selectOneOfTheseInstances.

LinkedList<SelectItem> getSelectOneOfTheseMetamodels ()

Definiert in Zeile 99 der Datei SelectObject.java.

Benutzt SelectObject.initMetamodels() und SelectObject.selectOneOfTheseMetamodels.

LinkedList<SelectItem> getSelectOneOfTheseObjects ()

Definiert in Zeile 195 der Datei SelectObject.java.

Benutzt SelectObject.selectOneOfTheseObjects.

void initInstances () [private]

Definiert in Zeile 225 der Datei SelectObject.java.

Benutzt SelectObject.password, SelectObject.selectOneOfTheseInstances und SelectObject.userId.

Wird benutzt von SelectObject.setSelectedObject().

void initMetamodels () [private]

Definiert in Zeile 108 der Datei SelectObject.java.

Benutzt SelectObject.categoryDAO und SelectObject.selectOneOfTheseMetamodels.

Wird benutzt von SelectObject.getSelectedOneOfTheseMetamodels().

void initObjects () [private]

Definiert in Zeile 141 der Datei SelectObject.java.

Benutzt SelectObject.eamObjectDAO, SelectObject.metamodel und SelectObject.selectOneOfTheseObjects.

Wird benutzt von SelectObject.setSelectedMetamodel().

String readTemplate (String path) [private]

Diese Methode liest die Datei unter dem gegebenen Pfad ein.

Parameter:

path Hier liegt die Textdatei.

Rückgabe:

Ein String, der den Inhalt der Textdatei enthält.

Definiert in Zeile 380 der Datei SelectObject.java.

Wird benutzt von SelectObject.analyse().

void selectInstance (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 258 der Datei SelectObject.java.

Benutzt SelectObject.instance, SelectObject.password, SelectObject.selectedInstance, SelectObject.setSelectedInstance() und SelectObject.userId.

void selectMetamodel (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 91 der Datei SelectObject.java.

Benutzt SelectObject.categoryDAO, SelectObject.metamodel, SelectObject.selectedMetamodel und SelectObject.setSelectedMetamodel().

void selectObject (ValueChangeEvent valueChangeEvent)

Definiert in Zeile 171 der Datei SelectObject.java.

Benutzt SelectObject.eamObjectDAO, SelectObject.eamRelationDAO, SelectObject.selectedObject und SelectObject.setSelectedObject().

void setCostTypeObject (String costTypeObject)

Definiert in Zeile 431 der Datei SelectObject.java.

void setMap (HashMap< String, String > map)

Definiert in Zeile 413 der Datei SelectObject.java.

void setPerson (String person)

Definiert in Zeile 419 der Datei SelectObject.java.

void setRelationPerson (String relationPerson)

Definiert in Zeile 425 der Datei SelectObject.java.

void setSelectedInstance (String instance)

Definiert in Zeile 219 der Datei SelectObject.java.

Benutzt SelectObject.selectedInstance.

Wird benutzt von SelectObject.selectInstance().

void setSelectedMetamodel (String metamodel)

Definiert in Zeile 75 der Datei SelectObject.java.

Benutzt SelectObject.initObjects(), SelectObject.instance, SelectObject.selectedInstance, SelectObject.selectedMetamodel, SelectObject.selectedObject, SelectObject.selectOneOfTheseInstances und SelectObject.selectOneOfTheseObjects.

Wird benutzt von SelectObject.selectMetamodel().

void setSelectedObject (String object)

Definiert in Zeile 131 der Datei SelectObject.java.

Benutzt SelectObject.initInstances() und SelectObject.selectedObject.

Wird benutzt von SelectObject.selectObject().

11.4.2.3 Dokumentation der Datenelemente

final EAMCategoryDAO categoryDAO = DAOFactory.getAuthInstance(userId, password).getEAMCategoryDAO() [private]

Definiert in Zeile 45 der Datei SelectObject.java.

Wird benutzt von SelectObject.initMetamodels() und SelectObject.selectMetamodel().

String costTypeObject = "server" [private]

Definiert in Zeile 66 der Datei SelectObject.java.

Wird benutzt von SelectObject.analyse() und SelectObject.getCostTypeObject().

final EAMObjectDAO eamObjectDAO = DAOFactory.getAuthInstance(userId, password).getEAMObjectDAO() [private]

Definiert in Zeile 46 der Datei SelectObject.java.

Wird benutzt von SelectObject.initObjects() und SelectObject.selectObject().

final EAMRelationDAO eamRelationDAO = DAOFactory.getAuthInstance(userId, password).getEAMRelationDAO() [private]

Definiert in Zeile 47 der Datei SelectObject.java.

Wird benutzt von SelectObject.selectObject().

DataObject instance = null [private]

Definiert in Zeile 58 der Datei SelectObject.java.

Wird benutzt von SelectObject.analyse(), SelectObject.selectInstance() und SelectObject.setSelectedMetamodel().

HashMap<String, String> map = new HashMap<String, String>() [private]

Definiert in Zeile 63 der Datei SelectObject.java.

Wird benutzt von SelectObject.analyse() und SelectObject.getMap().

EAMCategory metamodel = null [private]

Definiert in Zeile 50 der Datei SelectObject.java.

Wird benutzt von SelectObject.initObjects() und SelectObject.selectMetamodel().

EAMSuperClass object = null [private]

Definiert in Zeile 54 der Datei SelectObject.java.

**String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
[private]**

Definiert in Zeile 42 der Datei SelectObject.java.

Wird benutzt von SelectObject.initInstances() und SelectObject.selectInstance().

String person = "verantwortlicher" [private]

Definiert in Zeile 64 der Datei SelectObject.java.

Wird benutzt von SelectObject.analyse() und SelectObject.getPerson().

String relationPerson = "hat" [private]

Definiert in Zeile 65 der Datei SelectObject.java.

Wird benutzt von SelectObject.analyse() und SelectObject.getRelationPerson().

String selectedInstance = null [private]

Definiert in Zeile 59 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedInstance(), SelectObject.selectInstance(), SelectObject.setSelectedInstance() und SelectObject.setSelectedMetamodel().

String selectedMetamodel = null [private]

Definiert in Zeile 51 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedMetamodel(), SelectObject.selectMetamodel() und SelectObject.setSelectedMetamodel().

String selectedObject = null [private]

Definiert in Zeile 55 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedObject(), SelectObject.selectObject(), SelectObject.setSelectedMetamodel() und SelectObject.setSelectedObject().

**LinkedList<SelectItem> selectOneOfTheseInstances = new
LinkedList<SelectItem>() [private]**

Definiert in Zeile 60 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectOneOfTheseInstances(), SelectObject.initInstances() und SelectObject.setSelectedMetamodel().

**LinkedList<SelectItem> selectOneOfTheseMetamodels = new
LinkedList<SelectItem>() [private]**

Definiert in Zeile 52 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectOneOfTheseMetamodels() und SelectObject.initMetamodels().

```
LinkedList<SelectItem> selectOneOfTheseObjects = new  
LinkedList<SelectItem>() [private]
```

Definiert in Zeile 56 der Datei SelectObject.java.

Wird benutzt von SelectObject.getSelectedOneOfTheseObjects(), SelectObject.initObjects() und SelectObject.setSelectedMetamodel().

```
String userId =  
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")  
[private]
```

Definiert in Zeile 41 der Datei SelectObject.java.

Wird benutzt von SelectObject.initInstances() und SelectObject.selectInstance().

12 de.offis.pg.eam.mod_example

12.1 BundleConfiguration

12.1.1 BundleViews Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::module::api::IModuleView.

12.1.1.1 Ausführliche Beschreibung

Liest die Views des Bundles.

Definiert in Zeile 26 der Datei BundleViews.java.

Öffentliche Methoden

- String getBundleVersion ()
- String getDescription ()
- long getEquinoxBundleId ()
- String getName ()
- List< View > getViews ()

Statische private Attribute

- static final String VIEWS = "bundle_views.xml"

12.1.1.2 Dokumentation der Elementfunktionen

String getBundleVersion ()

Definiert in Zeile 72 der Datei BundleViews.java.

String getDescription ()

Definiert in Zeile 33 der Datei BundleViews.java.

long getEquinoxBundleId ()

Definiert in Zeile 38 der Datei BundleViews.java.

String getName ()

Definiert in Zeile 43 der Datei BundleViews.java.

List<View> getViews ()

Definiert in Zeile 48 der Datei BundleViews.java.

Benutzt BundleViews.VIEWS.

12.1.1.3 Dokumentation der Datenelemente

final String VIEWS = "bundle_views.xml" [static, private]

Definiert in Zeile 29 der Datei BundleViews.java.

Wird benutzt von BundleViews.getViews().

12.1.2 Configuration Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::configuration::api::IBundleConfiguration.

12.1.2.1 Ausführliche Beschreibung

Konfiguration des Bundles

Definiert in Zeile 16 der Datei Configuration.java.

Öffentliche Methoden

- boolean bundleHasObjects ()
- boolean bundleHasRelations ()
- boolean bundleHasViews ()
- Document getObjectsJdomDocument () throws JDOMException, IOException

12.1.2.2 Dokumentation der Elementfunktionen

boolean bundleHasObjects ()

Bringt das Bundle Objekte mit? Siehe TestBundleMatching

Definiert in Zeile 22 der Datei Configuration.java.

boolean bundleHasRelations ()

Bringt das Bundle Relationen mit? Siehe TestBundleMatching

Definiert in Zeile 46 der Datei Configuration.java.

boolean bundleHasViews ()

Bringt das Bundle Views mit?

Definiert in Zeile 30 der Datei Configuration.java.

Document getObjectsJdomDocument () throws JDOMException, IOException

Bringt das Bundle Objekte mit? Siehe TestBundleMatching

Definiert in Zeile 38 der Datei Configuration.java.

12.1.3 Activator Klassenreferenz

Abgeleitet von org::osgi::framework::BundleActivator.

12.1.3.1 Ausführliche Beschreibung

Definiert in Zeile 33 der Datei Activator.java.

Öffentliche Methoden

- void start (BundleContext context) throws Exception
- void stop (BundleContext context) throws Exception

Statische öffentliche Attribute

- static BundleContext bundleContext
- static String bundleSymbolicName
- static String bundleVersion
- static long equinoxBundleId
- static final String PATH = "/example"

Private Attribute

- ServiceTracker httpServiceTracker

Statische private Attribute

- static final String WEB_ROOT = "/WebRoot"

Klassen

- class HttpServiceTracker
- class ServletContextListenerServletAdaptor

12.1.3.2 Dokumentation der Elementfunktionen

void start (BundleContext context) throws Exception

Standard-Definitionen

Definiert in Zeile 51 der Datei Activator.java.

Benutzt Activator.bundleContext, Activator.bundleSymbolicName, Activator.bundleVersion, Activator.equinoxBundleId und Activator.httpServiceTracker.

void stop (BundleContext context) throws Exception

Definiert in Zeile 64 der Datei Activator.java.

Benutzt Activator.httpServiceTracker.

12.1.3.3 Dokumentation der Datenelemente

BundleContext bundleContext [static]

Definiert in Zeile 43 der Datei Activator.java.

Wird benutzt von Activator.start().

String bundleSymbolicName [static]

Definiert in Zeile 42 der Datei Activator.java.

Wird benutzt von Activator.start().

String bundleVersion [static]

Definiert in Zeile 44 der Datei Activator.java.

Wird benutzt von Activator.start().

long equinoxBundleId [static]

Definiert in Zeile 41 der Datei Activator.java.

Wird benutzt von `Activator.start()`.

ServiceTracker `httpServiceTracker` [private]

Definiert in Zeile 38 der Datei `Activator.java`.

Wird benutzt von `Activator.start()` und `Activator.stop()`.

final String `PATH = "/example"` [static]

Definiert in Zeile 36 der Datei `Activator.java`.

Wird benutzt von `Activator.HttpServiceTracker.addingService()` und `Activator.HttpServiceTracker.removedService()`.

final String `WEB_ROOT = "/WebRoot"` [static, private]

Definiert in Zeile 37 der Datei `Activator.java`.

Wird benutzt von `Activator.HttpServiceTracker.addingService()`.

12.1.4 Activator.HttpServiceTracker Klassenreferenz

Abgeleitet von `org::osgi::util::tracker::ServiceTracker`.

12.1.4.1 Ausführliche Beschreibung

Definiert in Zeile 68 der Datei `Activator.java`.

Öffentliche Methoden

- `Object addingService (ServiceReference reference)`
- `HttpServiceTracker (BundleContext context)`
- `void removedService (ServiceReference reference, Object service)`

12.1.4.2 Beschreibung der Konstruktoren und Destruktoren

HttpServiceTracker (BundleContext context)

Definiert in Zeile 70 der Datei `Activator.java`.

12.1.4.3 Dokumentation der Elementfunktionen

Object addingService (ServiceReference reference)

Definiert in Zeile 74 der Datei `Activator.java`.

Benutzt Activator.PATH und Activator.WEB_ROOT.

void removedService (ServiceReference reference, Object service)

Definiert in Zeile 124 der Datei Activator.java.

Benutzt Activator.PATH.

12.1.5 Activator.ServletContextListenerServletAdaptor Klassenreferenz

Abgeleitet von javax::servlet::Servlet.

12.1.5.1 Ausführliche Beschreibung

Definiert in Zeile 140 der Datei Activator.java.

Öffentliche Methoden

- void destroy ()
- ServletConfig getServletConfig ()
- String getServletInfo ()
- void init (ServletConfig config) throws ServletException
- void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException
- ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Private Attribute

- ServletConfig config
- Servlet delegate
- ClassLoader jspLoader
- ServletContextListener listener

12.1.5.2 Beschreibung der Konstruktoren und Destruktoren

ServletContextListenerServletAdaptor (ServletContextListener listener, Servlet delegate, ClassLoader jspLoader)

Definiert in Zeile 150 der Datei Activator.java.

12.1.5.3 Dokumentation der Elementfunktionen

void destroy ()

Definiert in Zeile 212 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config,
Activator.ServletContextListenerServletAdaptor.delegate,
Activator.ServletContextListenerServletAdaptor.jspLoader und
Activator.ServletContextListenerServletAdaptor.listener.

ServletConfig getServletConfig ()

Definiert in Zeile 238 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config.

String getServletInfo ()

Definiert in Zeile 244 der Datei Activator.java.

void init (ServletConfig config) throws ServletException

Definiert in Zeile 164 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.config,
Activator.ServletContextListenerServletAdaptor.delegate,
Activator.ServletContextListenerServletAdaptor.jspLoader und
Activator.ServletContextListenerServletAdaptor.listener.

void service (ServletRequest req, ServletResponse resp) throws ServletException, IOException

Definiert in Zeile 190 der Datei Activator.java.

Benutzt Activator.ServletContextListenerServletAdaptor.delegate und
Activator.ServletContextListenerServletAdaptor.jspLoader.

12.1.5.4 Dokumentation der Datenelemente

ServletConfig config [private]

Definiert in Zeile 142 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.getServletConfig() und
Activator.ServletContextListenerServletAdaptor.init().

Servlet delegate [private]

Definiert in Zeile 146 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.init() und
Activator.ServletContextListenerServletAdaptor.service().

ClassLoader jspLoader [private]

Definiert in Zeile 148 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy(),
Activator.ServletContextListenerServletAdaptor.init() und
Activator.ServletContextListenerServletAdaptor.service().

ServletContextListener listener [private]

Definiert in Zeile 144 der Datei Activator.java.

Wird benutzt von Activator.ServletContextListenerServletAdaptor.destroy() und
Activator.ServletContextListenerServletAdaptor.init().

12.2 Implementierung

12.2.1 ExampleBean Klassenreferenz

12.2.1.1 Ausführliche Beschreibung

Autor:

roko

Definiert in Zeile 28 der Datei ExampleBean.java.

Öffentliche Methoden

- DataObjectList getDataObjectList ()
- String getMetamodel ()
- LinkedList< SelectItem > getMetamodels ()
- String getObject ()
- LinkedList< SelectItem > getObjects ()
- void selectMetamodel (ValueChangeEvent valueChangeEvent)
- void selectObject (ValueChangeEvent valueChangeEvent)
- void setMessage (String msg)

Private Attribute

- DAOFactory daoFactory = DAOFactory.getAuthInstance(userId, password)
- IInstanceQuery instanceQuery
- EAMCategory metamodel = null
- EAMObject object = null
- String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
- String selectedMetamodel = ""
- String selectedObject = ""
- String userId =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")

12.2.1.2 Dokumentation der Elementfunktionen**DataObjectList getDataObjectList ()**

Liefer die DataObjectList zum aktuellen Objekt, für die der Benutzer entsprechende Zugriffsrechte besitzt.

Rückgabe:

DataObjectList

Definiert in Zeile 184 der Datei ExampleBean.java.

Benutzt ExampleBean.instanceQuery, ExampleBean.password, ExampleBean.setMessage() und ExampleBean.userId.

String getMetamodel ()

Definiert in Zeile 106 der Datei ExampleBean.java.

Benutzt ExampleBean.selectedMetamodel.

LinkedList<SelectItem> getMetamodels ()

Liefert alle Metamodelle, für die der Benutzer Zugriffsrechte besitzt.

Rückgabe:

LinkedList<EAMCategory>

Definiert in Zeile 74 der Datei ExampleBean.java.

Benutzt ExampleBean.daoFactory, ExampleBean.metamodel und ExampleBean.selectMetamodel().

String getObject ()

Definiert in Zeile 154 der Datei ExampleBean.java.

Benutzt ExampleBean.selectedObject.

LinkedList<SelectedItem> getObjects ()

Liefert alle Objekte des aktuellen Metamodells.

Rückgabe:

LinkedList<SelectedItem>

Definiert in Zeile 134 der Datei ExampleBean.java.

Benutzt ExampleBean.daoFactory, ExampleBean.metamodel und ExampleBean.selectObject().

void selectMetamodel (ValueChangeEvent valueChangeEvent)

valueChangeListener

Parameter:

valueChangeEvent

Definiert in Zeile 114 der Datei ExampleBean.java.

Benutzt ExampleBean.daoFactory, ExampleBean.metamodel und ExampleBean.selectedMetamodel.

Wird benutzt von ExampleBean.getMetamodels().

void selectObject (ValueChangeEvent valueChangeEvent)

valueChangeListener

Parameter:

valueChangeEvent

Definiert in Zeile 163 der Datei ExampleBean.java.

Benutzt ExampleBean.daoFactory und ExampleBean.selectedObject.

Wird benutzt von ExampleBean.getObjects().

void setMessage (String msg)

Definiert in Zeile 210 der Datei ExampleBean.java.

Wird benutzt von ExampleBean.getDataObjectList().

12.2.1.3 Dokumentation der Datenelemente

DAOFactory daoFactory = DAOFactory.getAuthInstance(userId, password)
[private]

Definiert in Zeile 38 der Datei ExampleBean.java.

Wird benutzt von ExampleBean.getMetamodels(), ExampleBean.getObjects(), ExampleBean.selectMetamodel() und ExampleBean.selectObject().

IInstanceQuery instanceQuery [private]

Initialisierung:

```
{
    try {
        instanceQuery = (IInstanceQuery)EAMProxy.newInstance(IInstanceQuery.class, Long.valueOf(userId), password);
    } catch (NumberFormatException e) {
        e.printStackTrace();
    } catch (ProxyException e) {
        e.printStackTrace();
    } catch (Throwable e) {
        e.printStackTrace();
    }
}
```

Definiert in Zeile 46 der Datei ExampleBean.java.

Wird benutzt von ExampleBean.getDataObjectList().

EAMCategory metamodel = null [private]

Definiert in Zeile 60 der Datei ExampleBean.java.

Wird benutzt von ExampleBean.getMetamodels(), ExampleBean.getObjects() und ExampleBean.selectMetamodel().

EAMObject object = null [private]

Definiert in Zeile 62 der Datei ExampleBean.java.

String password =
(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("password")
[private]

Definiert in Zeile 33 der Datei ExampleBean.java.

Wird benutzt von ExampleBean.getDataObjectList().

String selectedMetamodel = "" [private]

Definiert in Zeile 64 der Datei ExampleBean.java.

Wird benutzt von ExampleBean.getMetamodel() und ExampleBean.selectMetamodel().

String selectedObject = "" [private]

Definiert in Zeile 65 der Datei ExampleBean.java.

Wird benutzt von ExampleBean.getObject() und ExampleBean.selectObject().

String userId =

(String)FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user")
[private]

Definiert in Zeile 32 der Datei ExampleBean.java.

Wird benutzt von ExampleBean.getDataObjectList().

12.2.2 MenuConstants Schnittstellenreferenz

12.2.2.1 Ausführliche Beschreibung

Definiert in Zeile 3 der Datei MenuConstants.java.

Statische öffentliche Attribute

- static final String ADOPTER_ID = "exampleAdopter"
- static final String CONTRIBUTOR_ID = "exampleContributor"

12.2.2.2 Dokumentation der Datenelemente

final String ADOPTER_ID = "exampleAdopter" [static]

Definiert in Zeile 7 der Datei MenuConstants.java.

final String CONTRIBUTOR_ID = "exampleContributor" [static]

Definiert in Zeile 5 der Datei MenuConstants.java.

12.2.3 MenuContributor Klassenreferenz

Abgeleitet von de::offis::pg::eam::core::menu::api::AbstractAdoptableMenuContributor.

12.2.3.1 Ausführliche Beschreibung

Definiert in Zeile 6 der Datei MenuContributor.java.

Öffentliche Methoden

- String getAdopterId ()
- String getContributionTarget ()
- String getContributorId ()
- String getTargetSubmenu ()
- MenuContributor ()

12.2.3.2 Beschreibung der Konstruktoren und Destruktoren

MenuContributor ()

Definiert in Zeile 8 der Datei MenuContributor.java.

12.2.3.3 Dokumentation der Elementfunktionen

String getAdopterId ()

Definiert in Zeile 12 der Datei MenuContributor.java.

String getContributionTarget ()

Definiert in Zeile 16 der Datei MenuContributor.java.

String getContributorId ()

Definiert in Zeile 20 der Datei MenuContributor.java.

String getTargetSubmenu ()

Definiert in Zeile 24 der Datei MenuContributor.java.

12.2.4 DataObjectScrollerTag Klassenreferenz

12.2.4.1 Ausführliche Beschreibung

Das DataObjectScrollerTag stellt einen Scroller für dargestellte DataObjects durch das DataObjectTableTag dar.

Autor:

Roland Koppe

Definiert in Zeile 12 der Datei DataObjectScrollerTag.java.

Statische private Attribute

- static final long serialVersionUID = 1L

12.2.4.2 Dokumentation der Datenelemente

final long serialVersionUID = 1L [static, private]

Definiert in Zeile 14 der Datei DataObjectScrollerTag.java.

12.2.5 DataObjectTableTag Klassenreferenz

12.2.5.1 Ausführliche Beschreibung

Dieses Tag zeigt eine DataObjectList in einer Tabelle an.

Autor:

Roland Koppe

Definiert in Zeile 11 der Datei DataObjectTableTag.java.

Statische private Attribute

- static final long serialVersionUID = 1L

12.2.5.2 Dokumentation der Datenelemente

final long serialVersionUID = 1L [static, private]

Definiert in Zeile 13 der Datei DataObjectTableTag.java.