

Generated Documentation



Contents

Package com-makeabyte-agilephp Procedural Elements	2
AgilePHP.php	2
Function autoload	3
Function autoload class	3
Function autoload interceptions	3
Annotation.php	4
Crypto.php	5
Form.php	6
i18n.php	7
Identity.php	8
Interception.php	9
Logger.php	10
MVC.php	11
PersistenceManager.php	12
Remoting.php	13
Scope.php	14
Upload.php	15
Validator.php	16
Package com-makeabyte-agilephp Classes	17
Class AgilePHP	17
Method addInterception	17
Method disableInterceptors	17
Method enableInterceptors	18
Method getAppName	18
Method getDocumentRoot	18
Method getFramework	18
Method getFrameworkRoot	19
Method getInterceptions	19
Method getMVC	19
Method getRequestBase	19
Method getWebRoot	20
Method getXmlConfiguration	20
Method initMVC	20
Method isDebugEnabled	20
Method setAppName	21
Method setDebugEnabled	21
Method setDefaultTimezone	21
Method setDisplayPhpErrors	22
Method setDocumentRoot	22
Method setFrameworkRoot	22
Method setRequestBase	23
Method setWebRoot	23
Method startClock	23

Method stopClock	24
Method destruct	24
Class AgilePHP AccessDeniedException	24
Class AgilePHP AnnotationException	24
Class AgilePHP Exception	25
Constructor construct	25
Class AgilePHP InterceptionException	26
Class AgilePHP NotLoggedInException	26
Class AgilePHP PersistenceException	26
Class AgilePHP RemotingException	27
Constructor construct	27
Class Annotation	27
Method getClass	28
Method getClassAsArray	28
Method getMethod	29
Method getMethodsAsArray	29
Method getPropertiesAsArray	29
Method getProperty	30
Method hasAnnotations	30
Class Crypto	31
Constructor construct	31
Method crc32	31
Method createIV	32
Method decrypt 3des	32
Method decrypt aes256	32
Method decrypt blowfish	32
Method encrypt 3des	33
Method encrypt aes256	33
Method encrypt blowfish	34
Method getAlgorithm	34
Method getDigest	34
Method getInstance	35
Method getSupportedHashAlgorithms	35
Method hash	35
Method md5	35
Method setAlgorithm	36
Method sha1	36
Method sha256	37
Method sha384	37
Method sha512	37
Class Form	38
Constructor construct	38
Method getAction	38
Method getEnctype	39
Method getForeignKeySelection	39
Method getHTML	39
Method getId	39
Method getMode	40
Method getModel	40

Method getName	40
Method getRequestToken	40
Method getXSL	40
Method setAction	41
Method setEnctype	41
Method setId	42
Method setMode	42
Method setModel	42
Method setName	43
Method setRequestToken	43
Class i18n	43
Method getDomain	44
Method getInstance	44
Method getLocale	44
Method setDomain	44
Method setLocale	45
Method translate	45
Class Identity	46
Method delete	46
Method forgotPassword	46
Method getCreated	46
Method getEmail	47
Method getInstance	47
Method getLastLogin	47
Method getModel	47
Method getPassword	47
Method getPasswordResetUrl	48
Method getRole	48
Method getUsername	48
Method hasRole	48
Method isLoggedIn	49
Method login	49
Method logout	49
Method merge	50
Method persist	50
Method refresh	50
Method resetPassword	50
Method revokeRole	51
Method setCreated	51
Method setEmail	51
Method setLastLogin	52
Method setModel	52
Method setPassword	52
Method setPasswordResetUrl	53
Method setRole	53
Method setUsername	54
Method __destruct	54
Class Interception	54
Constructor __construct	55

Method createInterceptedTarget	55
Method createInterceptorProxy	55
Method getClass	55
Method getInterceptor	56
Method getMethod	56
Method getProperty	56
Method getSourceCode	56
Class Logger	57
Method debug	57
Method error	57
Method getInstance	58
Method info	58
Method setConfig	58
Method warn	59
Class MVC	59
Method createCustomRenderer	59
Method createDefaultRenderer	60
Method createRenderer	60
Method getAction	60
Method getController	61
Method getDefaultAction	61
Method getDefaultController	61
Method getDefaultRenderer	61
Method getInstance	61
Method processRequest	62
Method setConfig	62
Method setDefaultAction	62
Method setDefaultController	63
Method setDefaultRenderer	63
Class PersistenceManager	64
Constructor construct	64
Method beginTransaction	64
Method commit	65
Method compareModels	65
Method connect	65
Method create	66
Method createQuery	66
Method delete	66
Method drop	66
Method exec	67
Method execute	67
Method executeCountQuery	67
Method executeQuery	68
Method find	68
Method getColumnNameByProperty	68
Method getCount	69
Method getDistinct	69
Method getIdentityModel	69
Method getIdentityTable	69

Method getMaxResults	70
Method getModel	70
Method getModelName	70
Method getNextResultList	70
Method getOrderBy	70
Method getPage	71
Method getPageCount	71
Method getPreviousResultList	71
Method getPropertyNameByColumn	71
Method getQuery	72
Method getResultCount	72
Method getResultList	72
Method getSelectedDatabase	72
Method getSessionModel	73
Method getSessionTable	73
Method getTableByModel	73
Method getTableByModelName	73
Method getTableByName	74
Method getTableName	74
Method merge	74
Method nextExists	75
Method persist	75
Method prepare	75
Method previousExists	76
Method query	76
Method quote	76
Method rollBack	77
Method setDistinct	77
Method setGroupBy	77
Method setMaxResults	78
Method setModel	78
Method setOrderBy	78
Method setPage	79
Method setRestrictions	79
Method setRestrictionsLogicOperator	80
Method toAccessor	80
Method toBigInt	80
Method toMutator	81
Method truncate	81
Class Remoting	81
Constructor __construct	82
Method createInterceptedStub	82
Method createStandardStub	82
Method createStub	83
Method destroySession	83
Method getBaseJS	84
Method getClass	84
Method getSessionId	84
Method invoke	84

Method invokeIntercepted	85
Method invokeInterceptedStateful	85
Method invokeStateful	85
Method setClass	85
Class Scope	86
Method getApplicationScope	86
Method getInstance	87
Method getRequestScope	87
Method getSessionScope	87
Class Upload	87
Constructor construct	88
Method delete	88
Method getDirectory	88
Method getName	88
Method save	88
Method setDirectory	89
Method setName	89
Class Validator	89
Constructor construct	90
Method getInstance	90
Method validateEmail	90
Method validateIP	90
Method validateNumber	91
Method validateString	91

[Package com-makeabyte-agilephp-annotation Procedural](#)

Elements	94
AnnotatedClass.php	94
AnnotatedMethod.php	95
AnnotatedProperty.php	96
AnnotationParser.php	97

[Package com-makeabyte-agilephp-annotation Classes](#)

Class AnnotatedClass	98
Constructor construct	98
Method getAnnotation	99
Method getAnnotations	99
Method getMethod	99
Method getMethods	100
Method getProperties	100
Method getProperty	100
Method hasAnnotation	100
Method isAnnotated	101
Class AnnotatedMethod	101
Constructor construct	101
Method getAnnotation	102
Method getAnnotations	102
Method getDeclaringClass	102
Method hasAnnotation	103
Method isAnnotated	103

Class AnnotatedProperty	103
Constructor construct	104
Method getAnnotation	104
Method getAnnotations	104
Method getDeclaringClass	105
Method hasAnnotation	105
Method isAnnotated	105
Class AnnotationParser	106
Method getClassAnnotations	106
Method getClassAnnotationsToArray	106
Method getInstance	107
Method getMethodAnnotations	107
Method getMethodAnnotationsToArray	107
Method getPropertyAnnotations	107
Method getPropertyAnnotationsToArray	108
Method getSourceCode	108
Method parse	108
Package com-makeabyte-agilephp-annotation-annotations	
Procedural Elements	111
AroundInvoke.php	111
Interceptor.php	112
RemoteMethod.php	113
Package com-makeabyte-agilephp-annotation-annotations	
Classes	114
Class AroundInvoke	114
Class Interceptor	114
Class RemoteMethod	115
Package com-makeabyte-agilephp-identity Procedural Elements	117
IdentityManager.php	117
IdentityModel.php	118
Role.php	119
User.php	120
Package com-makeabyte-agilephp-identity Classes	121
Class IdentityManager	121
Method delete	121
Method forgotPassword	121
Method getCreated	122
Method getEmail	122
Method getLastLogin	122
Method getModel	122
Method getPassword	122
Method getPasswordResetUrl	123
Method getRole	123
Method getUsername	123
Method hasRole	123
Method isLoggedIn	124
Method login	124

Method logout	124
Method merge	124
Method persist	125
Method refresh	125
Method resetPassword	125
Method revokeRole	126
Method setCreated	126
Method setEmail	126
Method setLastLogin	126
Method setModel	127
Method setPassword	127
Method setPasswordResetUrl	127
Method setRole	128
Method setUsername	128
Class IdentityModel	129
Method getCreated	129
Method getEmail	129
Method getLastLogin	129
Method getPassword	129
Method getRole	130
Method getSessionId	130
Method getUsername	130
Method setCreated	130
Method setEmail	130
Method setLastLogin	131
Method setPassword	131
Method setRole	131
Method setSessionId	131
Method setUsername	132
Class Role	132
Constructor construct	132
Method getDescription	133
Method getName	133
Method setDescription	133
Method setName	133
Class User	134
Constructor User	134
Method getCreated	134
Method getEmail	134
Method getLastLogin	135
Method getPassword	135
Method getRole	135
Method getRoleId	135
Method getRoles	135
Method getSession	135
Method getSessionId	136
Method getUsername	136
Method setCreated	136
Method setEmail	136

Method setLastLogin	136
Method setPassword	137
Method setRole	137
Method setRoleId	137
Method setRoles	138
Method setSession	138
Method setSessionId	138
Method setUsername	138
Package com-makeabyte-agilephp-interception Procedural	
Elements	141
InterceptorProxy.php	141
InvocationContext.php	142
Package com-makeabyte-agilephp-interception Classes	143
Class InterceptorProxy	143
Constructor construct	143
Method getInstance	143
Method getInterceptedInstance	144
Method call	144
Method get	144
Method isset	145
Method set	145
Method unset	145
Class InvocationContext	146
Var \$proceed	146
Constructor construct	146
Method getInterceptor	147
Method getMethod	147
Method getParameters	147
Method getTarget	147
Method proceed	148
Method setMethod	148
Method setParameters	148
Package com-makeabyte-agilephp-interception-interceptors	
Procedural Elements	151
In.php	151
Restrict.php	152
Package com-makeabyte-agilephp-interception-interceptors	
Classes	153
Class In	153
Var \$class	153
Class Restrict	153
Var \$message	154
Var \$role	154
Var \$roles	154
Method requireRole	154
Package com-makeabyte-agilephp-mvc Procedural Elements	157

AJAXRenderer.php	157
BaseController.php	158
BaseModelActionController.php	159
BaseModelController.php	160
BaseModelXmlController.php	161
BaseModelXslController.php	162
BaseRenderer.php	163
PHTMLRenderer.php	164
XSLTRenderer.php	165
Package com-makeabyte-agilephp-mvc Classes	166
Class AJAXRenderer	166
Method render	166
Method renderNoFormat	167
Method renderNoFormatNoHeader	167
Method renderNoHeader	167
Method setOutput	168
Class BaseController	168
Constructor construct	168
Method createCustomRenderer	169
Method createRenderer	169
Method getBaseJS	169
Method getRenderer	170
Method index	170
Class BaseModelActionController	170
Constructor construct	171
Method delete	171
Method getXsltRenderer	171
Method merge	171
Method modelAction	172
Method modelList	172
Method modelSort	173
Method persist	173
Method revokeAccess	173
Class BaseModelController	174
Constructor construct	174
Method createQuery	174
Method delete	175
Method executeCountQuery	175
Method executeQuery	175
Method find	175
Method getCount	176
Method getMaxResults	176
Method getModel	176
Method getModelName	176
Method getNextResultList	176
Method getOrderBy	177
Method getPage	177
Method getPageCount	177
Method getPersistenceManager	177

Method <code>getPreviousResultList</code>	178
Method <code>getQuery</code>	178
Method <code>getResultCount</code>	178
Method <code>getResultList</code>	178
Method <code>getResultListAsModels</code>	178
Method <code>getTableName</code>	179
Method <code>merge</code>	179
Method <code>nextExists</code>	179
Method <code>persist</code>	180
Method <code>previousExists</code>	180
Method <code>setGroupBy</code>	180
Method <code>setMaxResults</code>	180
Method <code>setOrderBy</code>	181
Method <code>setPage</code>	181
Method <code>setRestrictions</code>	181
Method <code>setRestrictionsLogicOperator</code>	182
Method <code>toAccessor</code>	182
Method <code>toMutator</code>	183
Class <code>BaseModelXmlController</code>	183
Constructor <code>construct</code>	183
Method <code>getModelAsFormXML</code>	183
Method <code>getModelPersistenceAction</code>	184
Method <code>getResultListAsPagedXML</code>	184
Method <code>getResultListAsXML</code>	185
Class <code>BaseModelXslController</code>	185
Constructor <code>construct</code>	186
Method <code>getModelAsReadOnlyXSL</code>	186
Method <code>getModelFormXSL</code>	186
Method <code>getModelListXSL</code>	187
Method <code>getPaginationXSL</code>	187
Class <code>BaseRenderer</code>	187
Method <code>get</code>	188
Method <code>getStore</code>	188
Method <code>render</code>	188
Method <code>set</code>	189
Class <code>PHTMLRenderer</code>	189
Method <code>render</code>	189
Method <code>renderComponent</code>	190
Class <code>XSLTRenderer</code>	190
Method <code>ErrorHandler</code>	191
Method <code>getRenderedXslFile</code>	191
Method <code>render</code>	191
Method <code>renderXslFile</code>	192
Method <code>transform</code>	192

[Package `com-makeabyte-agilephp-persistence` Procedural](#)

Elements	195
BasePersistence.php	195
Column.php	197

Database.php	198
ForeignKey.php	199
PersistenceRenderer.php	200
Table.php	201
Package com-makeabyte-agilephp-persistence Classes	202
Class BasePersistence	202
Var \$database	202
Var \$model	202
Var \$pdo	203
Var \$transactionInProgress	203
Constructor __construct	203
Method beginTransaction	203
Method close	203
Method commit	204
Method compare	204
Method delete	204
Method exec	205
Method execute	205
Method find	205
Method getAvailableDrivers	206
Method getColumnNameForProperty	206
Method getDatabase	207
Method getDisplayNameForColumn	207
Method getDistinct	207
Method getIdentityModel	207
Method getIdentityTable	208
Method getMaxResults	208
Method getPDOInstance	208
Method getPropertyNameForColumn	208
Method getSessionModel	209
Method getSessionTable	209
Method getTableByModel	209
Method getTableByModelName	210
Method getTableByName	210
Method isEmpty	210
Method merge	211
Method persist	211
Method prepare	211
Method query	212
Method quote	212
Method rollBack	212
Method setDistinct	213
Method setMaxResults	213
Method toAccessor	214
Method toBigInt	214
Method toMutator	214
Method truncate	215
Method __destruct	215
Class Column	215

Constructor	construct	216
Method	getDefault	216
Method	getDescription	216
Method	getDisplay	216
Method	getForeignKey	217
Method	getLength	217
Method	getModelPropertyName	217
Method	getName	217
Method	getProperty	217
Method	getType	218
Method	getViewDisplayName	218
Method	hasForeignKey	218
Method	isAutoIncrement	218
Method	isBit	219
Method	isForeignKey	219
Method	isPrimaryKey	219
Method	isRequired	219
Method	isSelectable	220
Method	isSortable	220
Method	isVisible	220
Method	setAutoIncrement	220
Method	setDefault	221
Method	setDescription	221
Method	setDisplay	221
Method	setForeignKey	222
Method	setLength	222
Method	setName	222
Method	setPrimaryKey	223
Method	setProperty	223
Method	setRequired	223
Method	setSelectable	224
Method	setSortable	224
Method	setType	224
Method	setVisible	225
Class Database		225
Constructor	construct	226
Method	addTable	226
Method	getHostname	226
Method	getId	226
Method	getName	227
Method	getPassword	227
Method	getTables	227
Method	getType	227
Method	getUsername	227
Method	hasTables	228
Method	setHostname	228
Method	setId	228
Method	setName	229
Method	setPassword	229

Method setTables	229
Method setType	230
Method setUsername	230
Class ForeignKey	230
Constructor construct	231
Method getColumnInstance	231
Method getFkColumn	231
Method getFkTable	232
Method getName	232
Method getOnDelete	232
Method getOnUpdate	232
Method getReferencedColumn	232
Method getReferencedColumnInstance	233
Method getReferencedController	233
Method getReferencedTable	233
Method getReferencedTableInstance	233
Method getSelect	233
Method getSelectedColumnInstance	234
Method getType	234
Method setFkColumn	234
Method setFkTable	235
Method setName	235
Method setOnDelete	235
Method setOnUpdate	236
Method setReferencedColumn	236
Method setReferencedController	236
Method setReferencedTable	237
Method setSelect	237
Method setType	237
Class PersistenceRenderer	238
Method isBit	238
Method toFormElement	239
Class Table	239
Constructor construct	239
Method addColumn	240
Method getColumnNameByProperty	240
Method getColumns	240
Method getDescription	241
Method getDisplay	241
Method getDisplayNameByProperty	241
Method getForeignKeyColumns	241
Method getForeignKeyColumnsByKey	242
Method getModel	242
Method getModelInstance	242
Method getName	242
Method getPrimaryKeyColumns	243
Method getSelectableColumns	243
Method getValidate	243
Method getViewDisplayName	243

Method hasBlobColumn	243
Method hasColumns	244
Method hasForeignKey	244
Method hasForeignKeyReferences	244
Method isIdentity	244
Method isSession	245
Method isVisible	245
Method setColumns	245
Method setDescription	246
Method setDisplay	246
Method setIsIdentity	246
Method setIsSession	247
Method setModel	247
Method setName	247
Method setValidate	248
Package com-makeabyte-agilephp-persistence-dialect	
Procedural Elements	250
MySQLDialect.php	250
SQLDialect.php	251
SQLiteDialect.php	252
Package com-makeabyte-agilephp-persistence-dialect Classes	253
Class MySQLDialect	253
Constructor construct	253
Method create	253
Method drop	254
Class SQLDialect	254
Method create	254
Method drop	255
Class SQLiteDialect	255
Constructor construct	255
Method create	255
Method drop	256
Method truncate	256
Package com-makeabyte-agilephp-scope Procedural Elements	258
ApplicationScope.php	258
RequestScope.php	259
Session.php	260
SessionScope.php	261
Package com-makeabyte-agilephp-scope Classes	262
Class ApplicationScope	262
Method clear	262
Method destroy	262
Method get	263
Method getInstance	263
Method persist	263
Method set	264
Method destruct	264

Class RequestScope	264
Method createToken	265
Method get	265
Method getInstance	265
Method getParameters	265
Method getSanitized	266
Method invalidate	266
Method sanitize	266
Method set	267
Class Session	267
Constructor construct	267
Method getCreated	267
Method getData	268
Method getId	268
Method setCreated	268
Method setData	268
Method setId	269
Class SessionScope	269
Method clear	270
Method destroy	270
Method get	270
Method getInstance	270
Method getSession	271
Method getSessionId	271
Method isPersisted	271
Method persist	271
Method set	271
Method setSessionId	272
Method destruct	272

Appendices	273
Appendix A - Class Trees	274
com-makeabyte-agilephp-annotation	274
com-makeabyte-agilephp-annotation-annotations	274
com-makeabyte-agilephp	275
com-makeabyte-agilephp-persistence	277
com-makeabyte-agilephp-persistence-dialect	278
com-makeabyte-agilephp-interception-interceptors	279
com-makeabyte-agilephp-interception	279
com-makeabyte-agilephp-scope	279
com-makeabyte-agilephp-mvc	280
com-makeabyte-agilephp-identity	281
Appendix B - README/CHANGELOG/INSTALL	282
README	283
Appendix D - Todo List	285

Package com-makeabyte-agilephp

Procedural Elements

AgilePHP.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework
Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp

require_once ['Annotation.php'](#)*[line 584]*

If the class being loaded is annotated with any interceptors, an `InterceptorProxy` class is created for the requested class. The `InterceptorProxy` is a template used to create the dynamic proxy for the class being intercepted.

Note: Annotations are being gotten as arrays here instead of `Annotated*` objects because in order to create the dynamic proxy with the name of the requested class, we need to make sure that the class has not been loaded since there is no easy/elegant way to unload a PHP class once it has been loaded.

require_once ['Interception.php'](#)*[line 586]*

require_once ['annotation/annotations/AroundInvoke.php'](#) [line 587]

require_once ['Interception.php'](#) [line 585]

void function __autoload(\$class) [line 555]

Function Parameters:

- **\$class \$class** The class being lazy loaded

Responsible for 'lazy loading' classes. If agilephp.xml contains a valid <interceptors/> configuration, the loaded class is handed off to __autoload_interceptions.

void function __autoload_class(\$class) [line 650]

Function Parameters:

- **\$class \$class** The name of the class being loaded by __autoload

Lazy loads standard framework and web application classes that do not contain interceptor annotations.

void function __autoload_interceptions(\$class) [line 589]

Function Parameters:

- **\$class**

Annotation.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp

require_once ['annotation/AnnotatedProperty.php'](#) [line 29]

require_once ['annotation/AnnotatedMethod.php'](#) [line 28]

require_once ['annotation/AnnotatedClass.php'](#) [line 27]

require_once ['annotation/AnnotationParser.php'](#) [line 26]

Includes all annotation package dependancies

Crypto.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp

Form.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp

i18n.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp

Identity.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp

require_once ['identity/Role.php'](#)*[line 33]*

require_once ['identity/IdentityModel.php'](#)*[line 32]*

require_once ['identity/IdentityManager.php'](#)*[line 31]*

NOTE: PDO objects can *NOT* be serialized. Since this component

is stored in the HTTP session (serialized), this class can not extend any of the base MVC controllers that store an instance of PDO, nor can it store the PersistenceManager itself!

Includes all identity package dependancies

Interception.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp

require_once ['interception/InvocationContext.php'](#) [line 26]

require_once ['interception/InterceptorProxy.php'](#) [line 25]

Includes all interception package dependancies

Logger.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp

MVC.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp

PersistenceManager.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp

require_once '[persistence/BasePersistence.php](#)' *[line 26]*

require_once '[persistence/dialect/SQLDialect.php](#)' *[line 25]*

Includes base persistence dependencies

Remoting.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp

Scope.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp

require_once '[scope/SessionScope.php](#)[line 28]

require_once '[scope/Session.php](#)[line 27]

require_once '[scope/RequestScope.php](#)[line 26]

require_once '[scope/ApplicationScope.php](#)[line 25]

Includes all scope dependencies

Upload.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp

Validator.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp

Package com-makeabyte-agilephp Classes

Class AgilePHP

[line 31]

AgilePHP :: Core Framework Class

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc
- **Static**

void function AgilePHP::addInterception(\$interception) [line 300]

Function Parameters:

- [Interception](#) **\$interception** The interception instance to add to the stack

Adds an Interception to the interceptions stack

- **Access** public

void function AgilePHP::disableInterceptors() [line 332]

Disables interceptors.

- **Access** public

void function AgilePHP::enableInterceptors() [*line 322*]

Enables the use of interceptors.

- **Access** public

AgilePHP function AgilePHP::getAppName() [*line 218*]

Returns the name of the AgilePHP web application.

- **Access** public

The function AgilePHP::getDocumentRoot() [*line 175*]

Returns the relative path to the web application from the server's document root.

- **Access** public

An function AgilePHP::getFramework() [*line 82*]

Factory method which is used to instantiate a singleton instance of the AgilePHP framework.

- **Static**
- **Access** public

The function AgilePHP::getFrameworkRoot() [line 154]

Gets the full system path to the location where AgilePHP resides.

- **Access public**

Array function AgilePHP::getInterceptions() [line 312]

Returns an array of Interceptions which have been loaded into the framework

- **Access public**

void function AgilePHP::getMVC() [line 104]

Accessor method for Model-View-Control component. The MVC component this method returns is created as follows:

1) If an agilephp.xml file is present in the web app root and contains a valid 'mvc' configuration, the MVC component is created using the specified values when AgilePHP framework is instantiated by a call to 'getFramework'.

2) A call to 'initMVC' has already loaded the MVC component so this method will return the already created component.

3) No agilephp.xml file is present. This method will obtain a singleton instance, store it internally, and then return the stored instance.

- **Access public**

The function AgilePHP::getRequestBase() [line 187]

Returns the base action url used to communicate with the AgilePHP MVC component. Defaults to the name of the script which initializes the framework.

- **Access** public

The function AgilePHP::getWebRoot() [line 129]

Returns the fully qualified path to the base directory of the web application.

- **Access** public

The function AgilePHP::getXmlConfiguration() [line 278]

Returns the agilephp.xml configuration as a SimpleXMLElement.

- **Access** public

void function AgilePHP::initMVC() [line 247]

Initializes a default Model-View-Control component without reading agilephp.xml configuration.

- **Access** public

void function AgilePHP::isInDebugMode() [line 268]

Whether or not AgilePHP framework is running in debug mode.

- **Access** public

void function AgilePHP::setAppName(\$name) [line 208]

Function Parameters:

- *\$name* **\$name** The name of the AgilePHP application

Sets the name of the AgilePHP web application.

- **Access** public

void function AgilePHP::setDebugMode(\$boolean) [line 258]

Function Parameters:

- *\$boolean* **\$boolean** True for debug mode, false for production mode. Default is production.

Enables or disables AgilePHP framework debug mode.

- **Access** public

void function AgilePHP::setDefaultTimezone(\$timezone) [line 289]

Function Parameters:

- *\$timezone* **\$timezone** The timezone to use as default. (ie. America/New_York)

Calls PHP date_default_timezone_set function to set the current timezone.

- **Access** public

void function AgilePHP::setDisplayPhpErrors(\$bool) [line 230]

Function Parameters:

- *\$bool* **\$bool** True to turn on error reporting on (E_ALL)

By default PHP hides errors on production servers. Setting this to true enables PHP 'display_errors', sets 'error_reporting' to 'E_ALL'.

- **Access** public

void function AgilePHP::setDocumentRoot(\$path) [line 165]

Function Parameters:

- *\$path* **\$path** The document root path

Sets the relative path to the web application from the server's document root.

- **Access** public

void function AgilePHP::setFrameworkRoot(\$path) [line 141]

Function Parameters:

- *\$path* **\$path** The full system path to the location where AgilePHP framework resides.

Sets the full system path to the location of the AgilePHP framework. The given path is appended to the current php.ini include_path configuration.

- **Access** public

void function AgilePHP::setRequestBase(\$url) [line 197]

Function Parameters:

- *\$url* **\$url** The base url to be used to communicate with the AgilePHP MVC component.

Sets the base action url used to communicate with the AgilePHP MVC component.

- **Access** public

void function AgilePHP::setWebRoot(\$path) [line 118]

Function Parameters:

- *\$path* **\$path** The fully qualified path to the web application

Sets the fully qualified path to the base directory of the web application.

- **Access** public

void function AgilePHP::startClock() [line 425]

Starts a timer. Useful for measuring how long a particular operation takes.

- **Access** public

Time function AgilePHP::stopClock() [line 439]

Stops the timer and returns the elapsed time between startClock() and stopClock().

- **Access** public

void function AgilePHP::__destruct() [line 450]

- **Access** public

Class AgilePHP_AccessDeniedException

[line 498]

Thrown when a user attempts to access content which they do not have permission to view.

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Class AgilePHP_AnnotationException

[line 508]

Exceptions thrown by the Annotation package

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Class AgilePHP_Exception

[line 461]

Base AgilePHP exception class

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void* function AgilePHP_Exception::__construct(\$message, [\$code = 0]) [line 463]

Function Parameters:

- **\$message**
- **\$code**

- **Access** public

Class AgilePHP_InterceptionException

[line 518]

Exceptions thrown by the Interception package

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Class AgilePHP_NotLoggedInException

[line 487]

Thrown when a user is not logged in and requests secure content

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Class AgilePHP_PersistenceException

[line 477]

Exceptions thrown by the Persistence/ORM framework

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Class AgilePHP_RemotingException

[line 529]

Handles all remoting exceptions. Output is returned in JSON format with an application/json HTTP header.

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void* function AgilePHP_RemotingException::__construct(\$message) [line 537]

Function Parameters:

- *\$message* **\$message** The exception message

Deliver remoting exceptions in JSON format and halt execution.

- **Access** public

Class Annotation

[line 40]

AgilePHP :: Annotation Enables the use of annotations in PHP.

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

void function Annotation::getClass(\$class) [*line 52*]

Function Parameters:

- *\$class* **\$class** The class name or instance to inspect

Static factory method used to retrieve AnnotatedClass instances.

- **Throws** AgilePHP_AnnotationException
- **Static**
- **Access** public

Array function Annotation::getClassAsArray(\$class) [*line 116*]

Function Parameters:

- *\$class* **\$class** The class name or instance to inspect.

Returns an array of class level annotations for the specified class.

Tries to return a cached set of results first. If no annotations are found the specified class is then parsed and the new result is returned.

- **Throws** AgilePHP_AnnotationException
- **Static**
- **Access** public

AnnotatedMethod function *Annotation::getMethod(\$class, \$method)* [line 65]

Function Parameters:

- *\$class* **\$class** The class name or instance to inspect.
- *\$name* **\$method** The method name

Static factory method used to retrieve AnnotatedMethod instances.

- **Throws** AgilePHP_AnnotationException
- **Static**
- **Access** public

Array function *Annotation::getMethodsAsArray(\$class)* [line 134]

Function Parameters:

- *\$class* **\$class** The class name or instance to inspect

Returns an array of method level annotations for the specified class/method.

- **Throws** AgilePHP_AnnotationException
- **Static**
- **Access** public

Array function *Annotation::getPropertiesAsArray(\$class, 1)* [line 155]

Function Parameters:

- *\$property* **1** The property/field name to inspect
- *\$class* **\$class** The class name or instance to inspect

Returns an array of property level annotations for the specified class/property.

Tries to return a caches set of annotations first. If no annotations are found then the specified class is then parsed and the new result is returned.

- **Throws** AgilePHP_AnnotationException
- **Static**
- **Access** public

AnnotatedProperty function Annotation::getProperty(\$class, \$property) [line 78]

Function Parameters:

- *\$class* **\$class** The class name or instance to inspect
- *\$property* **\$property** The property name

Static factory method used to retrieve AnnotatedProperty instances.

- **Throws** AgilePHP_AnnotationException
- **Static**
- **Access** public

True function Annotation::hasAnnotations(\$class) [line 92]

Function Parameters:

- *\$class* **\$class** The class name or instance to inspect

Returns true if the specified class contains and class, method, or property level annotations.

- **Throws** AgilePHP_AnnotationException
- **Static**
- **Access** public

Class Crypto

[line 31]

AgilePHP :: Crypto Provides one way hashing and encryption

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc.

Constructor *void* function Crypto::__construct() [line 41]

Initializes the Crypto component with the hasing algorithm defined in agilephp.xml

- **Access** public

The function Crypto::crc32(\$data) [line 166]

Function Parameters:

- **\$data** **\$data** The data to hash

Returns an CRC32 hashed string.

- **Access** public

void function Crypto::createIV() [line 200]

- **Access public**

Plain function Crypto::decrypt_3des(\$iv, \$key, \$data) [line 225]

Function Parameters:

- *\$iv* **\$iv** The IV/salt
- *\$key* **\$key** The secret key used to encrypt the data
- *\$data* **\$data** The data to encrypt

Decrypts Triple DES data

- **Access public**

AES function Crypto::decrypt_aes256(\$iv, \$key, \$data) [line 275]

Function Parameters:

- *\$key* **\$iv** The secret key used to encrypt the data
- *\$data* **\$key** The data to encrypt
- **\$data**

Decrypts the specified data which was previously encrypted using AES 256

- **Access public**

Plain function Crypto::decrypt_blowfish(\$iv, \$key, \$data) [line 251]

Function Parameters:

- **\$iv \$iv** The IV/salt
- **\$key \$key** The secret key used to encrypt the data
- **\$data \$data** The data to encrypt

Decrypts a string previously encrypted with encrypt_blowfish

- **Access** public

Triple function Crypto::encrypt_3des(\$iv, \$key, \$data) [line 213]

Function Parameters:

- **\$iv \$iv** The IV/salt
- **\$key \$key** The secret key used to encrypt the data
- **\$data \$data** The data to encrypt

Encrypts the specified data using Triple DES.

- **Access** public

AES function Crypto::encrypt_aes256(\$iv, \$key, \$data) [line 263]

Function Parameters:

- **\$key \$iv** The secret key used to encrypt the data
- **\$data \$key** The data to encrypt
- **\$data**

Encrypts the specified data using AES 256 encryption

- **Access** public

Blowfish function Crypto::encrypt_blowfish(\$iv, \$key, \$data) [line 238]

Function Parameters:

- **\$iv \$iv** The IV/salt
- **\$key \$key** The secret key used to encrypt the data
- **\$data \$data** The data to encrypt

Encrypts the specified data using Blowfish

- **Access** public

The function Crypto::getAlgorithm() [line 87]

Returns the algorithm the Crypto component is configured to perform a hashing operation with using the 'getDigest()' method.

- **Access** public

The function Crypto::getDigest(\$data) [line 100]

Function Parameters:

- **\$data \$data** The data to hash

Returns the hashed \$data. This operation requires either a valid configuration in agilephp.xml for the Crypto component or you must manually set the algorithm with a call to 'setAlgorithm()'.

- **Access** public

Singleton function Crypto::getInstance() [line 54]

Returns a singleton instance of Crypto

- **Static**
- **Access** public

An function Crypto::getSupportedHashAlgorithms() [line 193]

Returns an array of supported hashing algorithms on the current PHP enabled web server.

- **Access** public

The function Crypto::hash(\$algorithm, \$data) [line 181]

Function Parameters:

- **The \$algorithm** algorithm to hash the defined data with. NOTE: You can get a list of supported algorithms on the server with a call to `getSupportedHashAlgorithms()`.
- **\$data \$data** The data to hash

Returns the hashed \$data parameter according to the defined \$algorithm parameter.

- **Access** public

The function Crypto::md5(\$data) [line 111]

Function Parameters:

- **\$data \$data** The data to hash

Returns a hashed MD5 string.

- **Access** public

void function Crypto::setAlgorithm(\$algorithm) [line 72]

Function Parameters:

- *\$algorithm* **\$algorithm** The algorithm to perform the hashing operation with. NOTE:
getSupportedHashAlgorithms() will return a list of algorithms available on the
server.

Sets the algorithm for the Crypto component for use with getDigest.

- **Throws** AgilePHP_Exception If passed a hashing name not available in
getSupportedHashAlgorithms().
- **Access** public

The function Crypto::sha1(\$data) [line 122]

Function Parameters:

- *\$data* **\$data** The data to hash

Returns an SHA1 hashed string.

- **Access** public

The function Crypto::sha256(\$data) [line 133]

Function Parameters:

- *\$data* **\$data** The data to hash

Returns an SHA256 hashed string.

- **Access** public

The function Crypto::sha384(\$data) [line 144]

Function Parameters:

- *\$data* **\$data** The data to hash

Returns an SHA384 hashed string.

- **Access** public

The function Crypto::sha512(\$data) [line 155]

Function Parameters:

- *\$data* **\$data** The data to hash

Returns an SHA512 hashed string.

- **Access** public

Class Form

[line 31]

AgilePHP :: Form Generates a form by reverse engineering the specified domain model.

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void* function Form::__construct([\$model = null], [\$id = null], [\$name = null], [\$action = null], [\$enctype = null], [\$request_token = null]) *[line 42]*

Function Parameters:

- **\$model**
- **\$id**
- **\$name**
- **\$action**
- **\$enctype**
- **\$request_token**

- **Access** public

The function Form::getAction() *[line 153]*

Returns the forms action attribute value

- **Access** public

The function `Form::getEnctype()` [line 174]

Returns the form's enctype attribute value

- **Access** public

The function `Form::getForeignKeySelection($foreignKey)` [line 330]

Function Parameters:

- [*ForeignKey*](#) `$foreignKey`

Returns an HTML select drop-down which contains a list of values for foreign key columns used in one-to-many relationships.

- **Access** public

The function `Form::getHTML()` [line 207]

Returns the form in HTML format

- **Access** public

The function `Form::getId()` [line 111]

Returns the form's id attribute value

- **Access** public

The function Form::getMode() [line 90]

Returns the form presentation mode.

- **Access** public

The function Form::getModel() [line 69]

Returns the domain model instance being used to create the form.

- **Access** public

The function Form::getName() [line 132]

Returns the form's HTML name attribute value

- **Access** public

anti-CSFR function Form::getRequestToken() [line 197]

Returns the anti-CSFR token

- **Access** public

XSL function Form::getXSL([\$keyValues = null], [\$page = 1]) [line 378]

Function Parameters:

- *\$pkeyValues* **\$pkeyValues** The primary key values for the ActiveRecord displayed in the form. Multiple primary keys must be separated by an underscore _ character. Default is null
- *\$page* **\$page** The list page number to return the user after the form is submitted. Default is 1.

Returns the form in XSL format

- **Access** public

void function Form::setAction(\$action) [*line 143*]

Function Parameters:

- *\$action* **\$action** The form's action

Sets the form action attribute value

- **Access** public

void function Form::setEnctype(\$enctype) [*line 164*]

Function Parameters:

- *\$enctype* **\$enctype** The form's enctype

Sets the form's enctype attribute value

- **Access** public

void function Form::setId(\$id) [line 101]

Function Parameters:

- ***\$id \$id*** The form's id

Sets the HTML id attribute value.

- **Access** public

void function Form::setMode(\$mode) [line 80]

Function Parameters:

- ***\$mode \$mode*** The presentation mode (persist|merge).

Sets the form presentation mode.

- **Access** public

void function Form::setModel(\$model) [line 59]

Function Parameters:

- ***\$model \$model*** The domain model object used to create the form

Sets the domain model that is used to create the form

- **Access** public

void function Form::setName(\$name) [*line 122*]

Function Parameters:

- *\$name* **\$name** The form's name

Sets the HTML name attribute value

- **Access** public

unknown_type function Form::setRequestToken(\$token) [*line 187*]

Function Parameters:

- *\$token* **\$token** CSFR token as supplied by RequestScope->getToken().

Sets the anti-CSFR token to render within the form. Note that the CSFR token should be different on every form!

- **See** [RequestScope::createToken\(\)](#)
- **Access** public

Class i18n

[*line 30*]

AgilePHP :: i18n (internationalization)

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

void function i18n::getDomain() [line 91]

Returns the messaging domain.

- **Access** public

Singleton function i18n::getInstance() [line 64]

Returns a singleton instance of i18n

- **Static**
- **Access** public

void function i18n::getLocale() [line 113]

Returns the language locale being used to translate

- **Access** public

void function i18n::setDomain(\$domain) [line 78]

Function Parameters:

- *\$domain* **\$domain** The messaging domain. Defaults to 'messages'.

Sets the messaging domain. This is the name of your .PO/.MO files.

- **Access** public

void function i18n::setLocale(\$locale) [line 102]

Function Parameters:

- *\$locale* **\$locale** The two letter_TWO LETTER language local. (ie. en_US, en_ES, ...)

Sets the language locale.

- **Access** public

void function i18n::translate(\$text) [line 124]

Function Parameters:

- *\$text* **\$text** The text to translate

Performs language translation based on the configured locale.

- **Static**
- **Access** public

Class Identity

[line 47]

AgilePHP :: Identity

Provides a means for tracking a users identity throughout the web application. The Identity component automatically handles creating sessions, persistence, and sending emails to deal with password resets.

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

void function Identity::delete() [line 449]
(non-PHPdoc)

- **See** delete()
- **Access** public

void function Identity::forgotPassword() [line 294]
(non-PHPdoc)

- **See** forgotPassword()
- **Access** public

void function Identity::getCreated() [line 210]
(non-PHPdoc)

- **See** getCreated()
- **Access** public

void function Identity::getEmail() [line 192]
(non-PHPdoc)

- **See** getEmail()
- **Access** public

void function Identity::getInstance() [line 113]
Returns a singleton instance of the 'Identity' component.

- **Static**
- **Access** public

void function Identity::getLastLogin() [line 228]
(non-PHPdoc)

- **See** getLastLogin()
- **Access** public

void function Identity::getModel() [line 134]
(non-PHPdoc)

- **See** getModel()
- **Access** public

void function Identity::getPassword() [line 174]

(non-PHPdoc)

- **See** getPassword()
- **Access** public

void function Identity::getPasswordResetUrl() [line 285]
(non-PHPdoc)

- **See** getPasswordResetUrl()
- **Access** public

void function Identity::getRole() [line 246]
(non-PHPdoc)

- **See** getRole()
- **Access** public

void function Identity::getUsername() [line 165]
(non-PHPdoc)

- **See** getUsername()
- **Access** public

void function Identity::hasRole(\$role) [line 255]
Function Parameters:

- **\$role**

(non-PHPdoc)

- **See** hasRole(\$role)
- **Access** public

void function Identity::isLoggedIn() [*line 420*]
(non-PHPdoc)

- **See** isLoggedIn()
- **Access** public

True function Identity::login(\$username, \$password) [*line 362*]
Function Parameters:

- *\$username* **\$username** The username to authenticate
- *\$password* **\$password** The password used to authenticate the account. The password is hashed using the Crypto component.

Authenticates the specified username and password.

- **See** authenticate()
- **Access** public

void function Identity::logout() [*line 410*]
(non-PHPdoc)

- **See** logout()
- **Access** public

void function Identity::merge() [line 439]
(non-PHPdoc)

- **See** merge()
- **Access** public

void function Identity::persist() [line 429]
(non-PHPdoc)

- **See** persist()
- **Access** public

void function Identity::refresh() [line 461]

Refreshes the IdentityManager domain object model's state by performing a fresh sql SELECT query.

- **Access** public

void function Identity::resetPassword(\$token, \$sessionId) [line 331]

Function Parameters:

- **\$token**
- **\$sessionId**

(non-PHPdoc)

- **See** resetPassword()
- **Access** public

void function Identity::revokeRole() [line 267]

(non-PHPdoc)

- **See** revokeRole(\$role)
- **Access** public

void function Identity::setCreated(\$dateTime) [line 201]

Function Parameters:

- **\$dateTime**

(non-PHPdoc)

- **See** setCreated(\$dateTime)
- **Access** public

void function Identity::setEmail(\$email) [line 183]

Function Parameters:

- **\$email**

(non-PHPdoc)

- **See** setEmail(\$email)
- **Access** public

void function Identity::setLastLogin(\$dateTime) [line 219]

Function Parameters:

- **\$dateTime**

(non-PHPdoc)

- **See** setLastLogin(\$dateTime)
- **Access** public

void function Identity::setModel(\$model) [line 125]

Function Parameters:

- **\$model**

(non-PHPdoc)

- **See** setModel(\$model)
- **Access** public

void function Identity::setPassword(\$password) [line 155]

Function Parameters:

- **\$password**

Sets encrypted/hashed password according to the configuration of the 'Crypto' component.
(non-PHPdoc)

- **See** setPassword(\$password)
- **Access** public

void function Identity::setPasswordResetUrl(\$url) [line 276]

Function Parameters:

- **\$url**

(non-PHPdoc)

- **See** setPasswordResetUrl(\$url)
- **Access** public

void function Identity::setRole(\$role) [line 237]

Function Parameters:

- [*Role*](#) **\$role**

(non-PHPdoc)

- **See** setRole(Role \$role)
- **Access** public

void function Identity::setUsername(\$username) [line 143]

Function Parameters:

- **\$username**

(non-PHPdoc)

- **See** setUsername(\$username)
- **Access** public

void function Identity::__destruct() [line 531]

Destructor

- **Access** public

Class Interception

[line 40]

AgilePHP :: Interception Performs interceptions by creating a dynamic proxy for intercepted classes. The proxy invokes the intended calls after inspecting (and/or intercepting) it according to the annotations in the intercepted object.

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc.

Constructor *void* function `Interception::__construct($class, $method, $property, $interceptor)` [line 54]

Function Parameters:

- **`$class`** **`$class`** The target class name
- **`$interceptor`** **`$method`** The instance of the interceptor which will intercept calls
- **`$property`**
- **`$interceptor`**

Initializes the Interception

- **Access** public

The function `Interception::createInterceptedTarget()` [line 108]

Creates a new intercepted target instance. The target is created by modifying the source code of the class being intercepted to `*classname*_Intercepted`.

- **Access** public

void function `Interception::createInterceptorProxy()` [line 126]

Creates and loads a dynamic proxy class which performs interceptions on the class created by `Interception::createInterceptedTarget()`.

- **Access** public

The function `Interception::getClass()` [line 70]

Returns the name of the target class being intercepted

- **Access** public

void function Interception::getInterceptor() [*line 97*]

Returns the interceptor instance which handles the intercepted call.

- **Access** public

void function Interception::getMethod() [*line 78*]

Returns the name of the target method to intercept

- **Access** public

void function Interception::getProperty() [*line 86*]

Returns the name of the property to intercept

- **Access** public

PHP function Interception::getSourceCode(\$class) [*line 158*]

Function Parameters:

- **\$class**

Returns the PHP file content to be parsed.

- **Throws** AgilePHP_InterceptionException if the source could not be loaded
- **Access** public

Class Logger

[line 31]

AgilePHP :: Logger Performs basic logging functionality

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

void function Logger::debug(\$message) [line 73]

Function Parameters:

- *\$message* **\$message** The debug message to log

Writes a 'debug' log level entry.

- **Access** public

void function Logger::error(\$message) [line 107]

Function Parameters:

- *\$message* **\$message** The error message to log.

Writes an 'error' log level entry.

- **Access** public

An function `Logger::getInstance()` [line 46]

Factory method that creates an instance of the configured Logger. If a logger has not been configured in agilephp.xml, an instance of 'FileLogger' is returned.

- **Static**
- **Access** public

void function `Logger::info($message)` [line 96]

Function Parameters:

- `$message` **\$message** The informative message to log

Writes an 'info' log level entry.

- **Access** public

void function `Logger::setConfig($config)` [line 61]

Function Parameters:

- `$config` **\$config** The SimpleXMLElement containing the agilephp.xml Logger configuration

Used by AgilePHP main framework class to automatically load Logger based on agilephp.xml configuration.

- **Access** public

void function Logger::warn(\$message) [line 85]

Function Parameters:

- *\$message* **\$message** The warning message to log

Writes a 'warn' log level entry.

- **Access** public

Class MVC

[line 30]

AgilePHP :: Model-View-Control (MVC)

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

A function MVC::createCustomRenderer(\$renderer, [\$classpath = ""]) [line 293]

Function Parameters:

- *\$renderer* **\$renderer** The name of the custom view renderer

- *\$classpath* **\$classpath** A relative child path under the webapp's 'classes' folder where the renderer is located.

Returns a new instance of the specified renderer. The renderer is loaded from the web app 'classes' directory.

- **Access** public

An function MVC::createDefaultRenderer() [*line 252*]

Returns a new instance of the default view renderer

- **Access** public

An function MVC::createRenderer(\$renderer) [*line 272*]

Function Parameters:

- **\$renderer**

Returns a new instance of the specified view renderer

- **Access** public

The function MVC::getAction() [*line 175*]

Returns the action currently being invoked.

- **Access** public

The function MVC::getController() [line 165]

Returns the name of the controller currently in use.

- **Access** public

The function MVC::getDefaultAction() [line 134]

**Returns the name of a default action method if one is not specified in the request URI.
Default is 'index'.**

- **Access** public

The function MVC::getDefaultController() [line 111]

**Returns the name of a default controller if one is not specified in the request URI.
Default is 'IndexController'.**

- **Access** public

The function MVC::getDefaultRender() [line 155]

Returns the name of the default view renderer

- **Access** public

Singleton function MVC::getInstance() [line 54]

Returns a singleton instance of MVC

- **Static**
- **Access** public

void function MVC::processRequest() [line 190]

Parses the current request URI to obtain the controller, action method, and arguments present for this request and then performs the invocation. If these parameters ARE NOT present, the default controller and default action method are used instead.

NOTE: The URI requirement to communicate with this MVC system is as follows
http://domain.com/ScriptName.php/ControllerName/ActionMethod/arg1/arg2/arg3/etc...

- **Access** public

void function MVC::setConfig(\$config) [line 69]

Function Parameters:

- **\$config \$config** SimpleXMLElement containing the MVC configuration.

Initializes the MVC component with agilephp.xml configuration.

- **Throws** AgilePHP_Exception If \$config is not an instance of SimpleXMLElement
- **Access** public

void function MVC::setDefaultAction(\$name) [line 123]

Function Parameters:

- **\$name \$name** The name of the default action method

Sets the name of the default action method if one is not specified in the request URI. Default is 'index'.

- **Access** public

void function MVC::setDefaultController(\$name) [line 100]

Function Parameters:

- **\$name \$name** The name of the controller

Sets the name of the default controller which is used if one is not specified in the request URI. Default is 'IndexController'.

- **Access** public

void function MVC::setDefaultRenderer(\$renderer) [line 145]

Function Parameters:

- **\$renderer \$renderer** The name of a view renderer to use as the default

Sets the name of the default view renderer. Default is 'PHTMLRenderer'.

- **Access** public

Class PersistenceManager

[line 37]

AgilePHP :: PersistenceManager Facade for working with persisence operations

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void* function PersistenceManager::__construct([\$databaseId = null]) [line 65]

Function Parameters:

- *\$databaseId* **\$databaseId** The id of the database to initalize PersistenceManager with. Defaults to the first database in persistence.xml.

Initializes the PersistenceManager object based on persistence.xml configuration. Each of the databases, tables, and columns are transformed into corresponding AgilePHP persistence objects.

If a 'databaseId' is present, PersistenceManager is initalized with the specified database.

- **Access** public

void function PersistenceManager::beginTransaction() [line 291]

Begins a transaction

- **See** <http://usphp.com/manual/en/function.pdo-beginTransaction.php>
- **See** <http://us2.php.net/manual/en/pdo.transactions.php>
- **Throws** AgilePHP_PersistenceException
- **Access** public

void function PersistenceManager::commit() [line 304]

Commits an already started transaction.

- See <http://usphp.com/manual/en/function.pdo-commit.php>
- See <http://us2.php.net/manual/en/pdo.transactions.php>
- **Throws** AgilePHP_PersistenceException
- **Access** public

True function PersistenceManager::compareModels(\$modelA, \$modelB) [line 183]

Function Parameters:

- *\$modelA* **\$modelA** The first domain model object
- *\$modelB* **\$modelB** The second domain model object

Checks the two defined model parameters to see if they are equal. The class name, property name, type and value.

- **Access** public

void function PersistenceManager::connect(\$db) [line 108]

Function Parameters:

- *\$db* **\$db** A Database object to establish the connection with

Establishes a connection to the specified database.

- **Access** public

void function PersistenceManager::create() [line 717]

Creates the database context as defined in persistence.xml.

- **Access** public

void function PersistenceManager::createQuery(\$sql) [line 389]

Function Parameters:

- *\$sql* **\$sql** A valid SQL statement

Sets the SQL statement to use when calling executeQuery.

- **Access** public

PDOException function PersistenceManager::delete(\$model) [line 764]

Function Parameters:

- *\$model* **\$model** The domain model object to delete

Deletes a persisted domain model object

- **Throws** AgilePHP_PersistenceException
- **Access** public

void function PersistenceManager::drop() [line 728]

Drops the database specified in persistence.xml

- **Throws** AgilePHP_PersistenceException
- **Access** public

The function PersistenceManager::exec(\$statement) [line 355]

Function Parameters:

- *\$statement* **\$statement** The SQL statement to execute.

Executes an SQL statement and returns the number of rows affected by the query.

- **See** <http://usphp.com/manual/en/function.pdo-exec.php>
- **Access** public

True function PersistenceManager::execute([\$inputParameters = array()]) [line 343]

Function Parameters:

- *\$inputParameters* **\$inputParameters** An array of input parameters

Executes a prepared statement (with parameters)

- **See** <http://usphp.com/manual/en/function.pdo-statement-execute.php>
- **Access** public

void function PersistenceManager::executeCountQuery() [line 802]

Executes an SQL count query for total number of records in the database.

Initializes 'pageCount' and 'count' properties.

- **Access** public

void function PersistenceManager::executeQuery() [*line 823*]

Executes the current SQL statement as defined in 'sql'.

- **Access** public

Returns function PersistenceManager::find(\$model, [\$findAll = false]) [*line 791*]

Function Parameters:

- *\$model* **\$model** A domain model object with its primary key field set
- *\$findAll* **\$findAll** True/false flag indicating whether or not to perform a SELECT * for empty model.

Attempts to locate the specified model by primary key value.

- **Throws** AgilePHP_PersistenceException
- **Access** public

The function PersistenceManager::getColumnByNameByProperty(\$table, \$property) [*line 658*]

Function Parameters:

- *\$table* **\$table** The 'Table' object to search
- *\$property* **\$property** The property attributes value

Returns the column 'name' attribute value configured in persistence.xml for the specified column 'property' attribute.

- **Access** public

Total function PersistenceManager::getCount() [*line 492*]

Returns the total number of records in the database table

- **Access** public

The function PersistenceManager::getDistinct() [*line 235*]

Returns the 'distinct' column to use in an SQL SELECT statement if one has been defined.

- **Access** public

The function PersistenceManager::getIdentityModel() [*line 623*]

Returns the domain object model responsible for 'Identity' persistence.

- **Access** public

The function PersistenceManager::getIdentityTable() [*line 613*]

Returns a SimpleXMLElement representing the table configured in persistence.xml as an AgilePHP 'Identity' table.

- **Access** public

The function PersistenceManager::getMaxResults() [line 278]

Returns the maximum number of results to retrieve. This translates to an SQL LIMIT clause during SELECT operations.

- **Access public**

The function PersistenceManager::getModel() [line 159]

Returns the domain model which the PersistenceManager is manipulating.

- **Access public**

The function PersistenceManager::getModelName() [line 194]

Returns the model's class name.

- **Access public**

void function PersistenceManager::getNextResultList() [line 545]

Sets the result list up with a 'next page' of a pagination request

- **Access public**

An function PersistenceManager::getOrderBy() [line 482]

Returns an associative array containing the current 'orderBy' clause. The results are returned with the name of the column as the index and the direction as the value.

- **Access** public

The function PersistenceManager::getPage() [line 502]

Returns the current page number

- **Access** public

The function PersistenceManager::getPageCount() [line 513]

Returns the total number of pages. This is calculated by dividing the total number of records by 'maxResults'.

- **Access** public

void function PersistenceManager::getPreviousResultList() [line 555]

Sets the result list up with a 'previous page' of a pagination request

- **Access** public

The function PersistenceManager::getPropertyNameByColumn(\$table, \$columnName) [line 671]

Function Parameters:

- **\$table** **\$table** The 'Table' object to search
- **\$columnName** **\$columnName** The column name to search

Returns the 'property' name configured in persistence.xml for the specified column 'name' attribute.

- **Access** public

The function PersistenceManager::getQuery() [line 399]

Returns the current SQL query

- **Access** public

The function PersistenceManager::getResultCount() [line 245]

Returns the total number of records in the current result list.

- **Access** public

An function PersistenceManager::getResultList() [line 256]

Returns a result set from the database. The result list is returned as an array of stdClass objects, each representing a row in the database.

- **Access** public

The function PersistenceManager::getSelectedDatabase() [line 565]

Returns the database currently being used by the PersistenceManager.

- **Access** public

void function PersistenceManager::getSessionModel() [line 645]

Returns the domain model object responsible for 'SessionScope' sessions.

- **Access** public

The function PersistenceManager::getSessionTable() [line 635]

Returns a SimpleXMLElement representing the table configured in persistence.xml as an AgilePHP 'SessionScope' session table.

- **Access** public

The function PersistenceManager::getTableByModel(\$model) [line 578]

Function Parameters:

- **\$model** **\$model** The domain model object to retrieve the table element for. Defaults to the model currently being managed by the 'PersistenceManager'.

Returns the 'Table' object which is mapped to the specified \$model.

- **Access** public

The function PersistenceManager::getTableByModelName(\$modelName) [line 590]

Function Parameters:

- **\$modelName**

Returns a 'Table' object representing the table configured in persistence.xml as the AgilePHP 'Identity' table.

- **Access** public

The function PersistenceManager::getTableByName(\$tableName) [*line 601*]

Function Parameters:

- *\$tableName* **\$tableName** The value of the table's 'name' attribute

Returns a 'Table' object by its name as configured in persistence.xml

- **Access** public

The function PersistenceManager::getTableName() [*line 210*]

Returns the table name for the model defined in the extension class

- **Access** public

PDOStatement function PersistenceManager::merge(\$model) [*line 752*]

Function Parameters:

- *\$model* **\$model** The model object to merge/update

Merges/updates a persisted domain model object

- **Throws** AgilePHP_PersistenceException
- **Access** public

True function PersistenceManager::nextExists() [*line 524*]

Returns boolean result based on whether or not a 'next page' result is available during a pagination request.

- **Access** public

PDOStatement function PersistenceManager::persist(\$model) [*line 740*]

Function Parameters:

- *\$model* **\$model** The model object to persist

Persists a domain model object

- **Throws** AgilePHP_PersistenceException
- **Access** public

False function PersistenceManager::prepare(\$statement) [*line 331*]

Function Parameters:

- *\$statement* **\$statement** The SQL statement to prepare

Prepares an SQL prepared statement

- See <http://usphp.com/manual/en/function.PDO-prepare.php>
- Access public

True function PersistenceManager::previousExists() [*line 535*]

Returns boolean result based on whether or not a 'previous page' result is available during a pagination request.

- Access public

PDO::PDOStatement function PersistenceManager::query(\$sql) [*line 379*]

Function Parameters:

- *\$sql* **\$sql** The SQL statement to execute

Executes a raw SQL query using PDO::query

- Access public

The function PersistenceManager::quote(\$data) [*line 368*]

Function Parameters:

- *\$data* **\$data** The data to quote

Quotes a string so its theoretically safe to pass into a statement without the worry of SQL injection.

- See <http://www.php.net/manual/en/pdo.quote.php>
- Access public

void function PersistenceManager::rollBack([\$message = null], [\$code = 0]) [*line 319*]

Function Parameters:

- *\$message* **\$message** Error/reason why the transaction was rolled back
- *\$code* **\$code** An error/reason code

Rolls back a transaction.

- See <http://usphp.com/manual/en/function.PDO-rollBack.php>
- See <http://us2.php.net/manual/en/pdo.transactions.php>
- Throws AgilePHP_PersistenceException
- Access public

void function PersistenceManager::setDistinct(\$columnName) [*line 224*]

Function Parameters:

- *\$columnName* **\$columnName** The column name to get the distinct values for

Adds an SQL distinct clause to 'find' operation.

- Access public

void function PersistenceManager::setGroupBy(\$column) [*line 457*]

Function Parameters:

- *\$column* **\$column** The column name to group the result set by

Sets the SQL 'group by' clause.

- **Access** public

void function PersistenceManager::setMaxResults(\$count) [line 267]

Function Parameters:

- *\$count* **\$count** Maximum number of records to return

Sets the maximum number of records to return in a result list

- **Access** public

void function PersistenceManager::setModel(\$model) [line 170]

Function Parameters:

- *\$model* **\$model** The domain model the PersistenceManager is to manipulate

Sets the domain model the PersistenceManager is to manipulate.

- **Access** public

void function PersistenceManager::setOrderBy(\$column, \$direction) [line 469]

Function Parameters:

- *\$column* **\$column** The column name to order the result set by *\$param* *\$direction* The direction to sort the result set (ASC|DESC).

- **\$direction**

Sets the SQL 'order by' clause.

- **Access** public

void function PersistenceManager::setPage(\$pageNumber) [line 411]

Function Parameters:

- *\$pageNumber* **\$pageNumber** The page number

Sets the pagination page number and performs an SQL query to populate the 'resultList' and 'resultCount' properties with their appropriate values for the specified page.

- **Access** public

void function PersistenceManager::setRestrictions(\$restrictions) [line 432]

Function Parameters:

- *\$restrictions* **\$restrictions** An associative array containing WHERE clause restrictions. (For example: array('id' => 21))

Sets WHERE clause restrictions

- **Access** public

void function PersistenceManager::setRestrictionsLogicOperator(\$operator) [line 443]

Function Parameters:

- *\$operator* **\$operator** The logical operator 'and'/'or' to be used in SQL WHERE clause. Default is 'AND'.

Sets the restriction operator (and|or) used in SQL WHERE clause.

- **Access** public

The function PersistenceManager::toAccessor(\$property) [line 684]

Function Parameters:

- *\$property* **\$property** The name of the property to convert to an accessor method name

Creates an accessor method from the \$property parameter. The \$property will be returned with the prefix 'get' and the first letter of the property uppercased.

- **Access** public

void function PersistenceManager::toBigInt(\$number) [line 707]

Function Parameters:

- **\$number**

Returns the specified number as a 'bigint' 64-bit number.

- **Access** public

The function PersistenceManager::toMutator(\$property) [line 697]

Function Parameters:

- *\$property* **\$property** The name of the property to convert to a mutator method name

Creates a mutator method from the \$property parameter. The \$property will be returned with the prefix 'set' and the first letter of the property uppercased.

- **Access** public

PDOStatement function PersistenceManager::truncate(\$model) [line 776]

Function Parameters:

- *\$model* **\$model** A domain model object

Truncates the table for the specified domain model object

- **Throws** AgilePHP_PersistenceException
- **Access** public

Class Remoting

[line 33]

AgilePHP :: Remoting Provides client side JavaScript remoting to PHP objects. Handles marshalling/unmarshalling of JSON objects between the client and server.

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc
- **Abstract Element**

Constructor *void* function Remoting::__construct([\$class = null]) [*line 37*]

Function Parameters:

- **\$class**

- **Access** protected

void function Remoting::createInterceptedStub([\$stateful = false]) [*line 307*]

Function Parameters:

- **\$stateful** **\$stateful** True to configure the remoting stub to invoke stateful server side calls. The remoted object is kept in the SessionScope.

Creates a dynamic javascript proxy stub/interface used for remoting AgilePHP intercepted classes

- **Throws** AgilePHP_RemotingException
- **Access** protected

void function Remoting::createStandardStub([\$stateful = false]) [*line 384*]

Function Parameters:

- **\$stateful \$stateful** True to configure the remoting stub to invoke stateful server side calls. The remotd object is kept in the SessionScope.

Creates a dynamic javascript proxy stub/interface used for remoting standard PHP classes

- **Throws** AgilePHP_RemotingException
- **Access** protected

void function Remoting::createStub([\$stateful = false]) [line 286]

Function Parameters:

- **\$stateful \$stateful** True to configure the remoting stub to invoke stateful server side calls. The remotd object is kept in the SessionScope.

Creates a dynamic javascript proxy stub/interface used for remoting PHP classes. This method handles both standard and intercepted classes.

- **Throws** AgilePHP_RemotingException
- **Access** protected

void function Remoting::destroySession(\$sessionId) [line 144]

Function Parameters:

- **\$sessionId \$sessionId** The id of the session to destroy

Destroys the session used for stateful remoting

- **Access** public

void function Remoting::getBaseJS([\$debug = false]) [line 456]

Function Parameters:

- **\$debug** **\$debug** True to enable client side AgilePHP debugging.

Returns the raw JavaScript contents of the AgilePHP.js file and pre-configures the library with a default AgilePHP.debug and AgilePHP.Remoting.controller value.

- **Access** public

The function Remoting::getClass() [line 57]

Returns the name of the class being remoted

- **Access** protected

Session function Remoting::getSessionId() [line 68]

Returns the current session id. If a session is not active a new session is created and the id is returned.

- **Access** public

Returns function Remoting::invoke() [line 158]

Invokes a non-persistent/stateful instance of the requested class/method passing in arguments if any were defined. This non-stateful approach is how most RPC web services work.

- **Throws** AgilePHP_RemotingException
- **Access** public

Returns function Remoting::invokeIntercepted() [line 189]

Performs RMI invocation on an intercepted class (non-stateful).

- **Throws** AgilePHP_RemotingException
- **Access** public

Returns function Remoting::invokeInterceptedStateful() [line 222]

Invokes the specified intercepted class/method/args using a stateful instance stored in SessionScope. If a stateful instance does not exist, a new instance is created and stored in the SessionScope for future calls.

- **Throws** AgilePHP_RemotingException
- **Access** public

Returns function Remoting::invokeStateful() [line 85]

Invokes the specified class/method/args using a stateful instance stored in SessionScope. If a stateful instance does not exist, a new instance is created and stored in the SessionScope for future calls.

- **Throws** AgilePHP_RemotingException
- **Access** public

void function Remoting::setClass(\$class) [line 48]

Function Parameters:

- **\$class \$class** The class name to remote

Sets the name of the class to remote

- **Access** protected

Class Scope

[line 39]

AgilePHP :: Scope Facade for AgilePHP scopes.

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

A function Scope::getApplicationScope([\$appName = null]) *[line 66]*

Function Parameters:

- **\$appName \$appName** An optional application name. Defaults to the HTTP HOST header.

Returns a singleton instance of ApplicationScope.

- **Access** public

An function `Scope::getInstance()` *[line 51]*

Returns a singleton instance of Scope.

- **Static**
- **Access** public

A function `Scope::getRequestScope()` *[line 90]*

Returns a singleton instance of RequestScope.

- **Access** public

void function `Scope::getSessionScope()` *[line 78]*

Returns a singleton instance of RequestScope.

- **Access** public

Class Upload

[line 31]

AgilePHP :: Upload File upload component

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a

- **Copyright** Make A Byte, inc

Constructor *void* function Upload::__construct() [*line 36*]

- **Access** public

void function Upload::delete() [*line 143*]

Deletes the uploaded file.

- **Access** public

void function Upload::getDirectory() [*line 75*]

Returns the destination directory for uploads.

- **Access** public

The function Upload::getName() [*line 54*]

Returns the file form name.

- **Access** public

The function Upload::save() [*line 87*]

Saves the upload contained in the \$_FILES array for the specified file input \$name.

- **Throws** AgilePHP_Exception if any errors are encountered.
- **Access** public

void function Upload::setDirectory(\$directory) [*line 65*]

Function Parameters:

- *\$directory* **\$directory** The upload destination directory

Sets the destination directory for the upload.

- **Access** public

void function Upload::setName(\$name) [*line 44*]

Function Parameters:

- *\$name* **\$name** The name attribute of the HTML file input element

Sets the name used in the file form field element.

- **Access** public

Class Validator

[*line 31*]

AgilePHP :: Validator Provides basic data validation

- **Package** com-makeabyte-agilephp
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void* function Validator::__construct() [*line 35*]

- **Access** public

An function Validator::getInstance() [*line 43*]

Factory method which returns a singleton instance of 'Validator'.

- **Throws** ReflectionException if the specified validator does not exist.
- **Static**
- **Access** public

True function Validator::validateEmail(\$email) [*line 58*]

Function Parameters:

- *\$email* **\$email** The email address to validate

Validates an email address by checking its syntax and performing and DNS lookup on the domain.

- **Static**
- **Access** public

True function Validator::validateIP(\$data) [*line 160*]

Function Parameters:

- **\$data \$data** The data to validate

Validates the specified data by ensuring it is a valid IP address.

- **Static**
- **Access** public

True function Validator::validateNumber(\$data) [*line 124*]

Function Parameters:

- **\$data \$data** The data to validate

Validates a number by ensuring it is either an int or float.

- **Static**
- **Access** public

True function Validator::validateString(\$data, [\$length = null]) [*line 140*]

Function Parameters:

- **\$data \$data** The data to validate
- **\$length \$length** Optional length parameter. If present the length is compared against the passed data to ensure its length is less than or equal to the specified \$length.

Validates the specified data by ensuring it is a string and its length is less than or equal to that of the specified \$length.

- **Static**
- **Access** public

Package com-makeabyte-agilephp-annotation Procedural Elements

AnnotatedClass.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework
Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-annotation

AnnotatedMethod.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-annotation

AnnotatedProperty.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-annotation

AnnotationParser.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-annotation

Package com-makeabyte-agilephp-annotation Classes

Class AnnotatedClass

[line 32]

AgilePHP :: AnnotatedClass Extends the PHP ReflectionClass to provide details about class level annotations.

- **Package** com-makeabyte-agilephp-annotation
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc.

Constructor *void* function AnnotatedClass::__construct(\$class) [line 42]

Function Parameters:

- *\$class* **\$class** The name or instance of the class to inspect

Initializes the AnnotatedClass by parsing the class file for annotations.

- **Access** public

The function `AnnotatedClass::getAnnotation($annotation)` [line 126]

Function Parameters:

- **\$name \$annotation** The name of the annotation

Gets an annotation instance by name. If the named annotation is found more than once, an array of annotations are returned.

- **Access** public

An function `AnnotatedClass::getAnnotations([$name = null])` [line 99]

Function Parameters:

- **\$name \$name** The name of the annotation to filter on. Default is null.

Returns all class level annotations. If a name is specified only annotations which match the specified name will be returned, otherwise all annotations are returned.

- **Access** public

`AnnotatedMethod` function `AnnotatedClass::getMethod($name)` [line 148]

Function Parameters:

- **\$name \$name** The method name

Returns an `AnnotatedMethod` instance for the specified method name.

- **Access** public

Array function AnnotatedClass::getMethods() [*line 159*]

Returns an array of AnnotatedMethod objects, one for each method in the class which contains an annotation.

- **Access** public

Array function AnnotatedClass::getProperties() [*line 189*]

Returns an array of AnnotatedProperty objects; one for each property in the class which contains an annotation.

- **Access** public

AnnotatedProperty function AnnotatedClass::getProperty(\$name) [*line 178*]

Function Parameters:

- *\$name* **\$name** The name of the property

Returns an AnnotatedProperty instance for the specified property name.

- **Access** public

True function AnnotatedClass::hasAnnotation(\$annotation) [*line 76*]

Function Parameters:

- *\$annotation* **\$annotation** The name of the annotation.

Checks the class for the presence of the specified annotation.

- **Access** public

True function AnnotatedClass::isAnnotated() *[line 65]*

Returns boolean indicator based on whether or not there are any annotations present.

- **Access** public

Class AnnotatedMethod

[line 32]

AgilePHP :: AnnotatedMethod Extends the PHP ReflectionClass to provide details about method level annotations.

- **Package** com-makeabyte-agilephp-annotation
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void* function AnnotatedMethod::__construct(\$class, \$method) *[line 36]*

Function Parameters:

- **\$class**
- **\$method**

- **Access** public

The function AnnotatedMethod::getAnnotation(\$annotation) [line 119]

Function Parameters:

- **\$name \$annotation** The name of the annotation

Gets an annotation instance by name. If the named annotation is found more than once, an array of annotations are returned.

- **Access** public

An function AnnotatedMethod::getAnnotations([\$name = null]) [line 93]

Function Parameters:

- **\$name \$name** The name of the annotation to filter on. Default is null.

Returns all method annotations. If a name is specified only annotations which match the specified name will be returned, otherwise all annotations are returned.

- **Access** public

AnnotatedClass function AnnotatedMethod::getDeclaringClass() [line 140]

Gets the parent class as an AnnotatedClass

- **Access** public

True function AnnotatedMethod::hasAnnotation(\$annotation) [line 69]

Function Parameters:

- *\$annotation* **\$annotation** The name of the annotation.

Checks the method for the presence of the specified annotation.

- **Access** public

True function AnnotatedMethod::isAnnotated() [line 58]

Returns boolean indicator based on the presence of any annotations.

- **Access** public

Class AnnotatedProperty

[line 32]

AgilePHP :: AnnotatedProperty Extends the PHP ReflectionClass to provide details about property level annotations.

- **Package** com-makeabyte-agilephp-annotation
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void* function AnnotatedProperty::__construct(\$class, \$property) [line 36]

Function Parameters:

- **\$class**
- **\$property**

- **Access** public

The function AnnotatedProperty::getAnnotation(\$annotation) [line 116]

Function Parameters:

- **\$name \$annotation** The name of the annotation

Gets an annotation instance by name. If the named annotation is found more than once, an array of annotations are returned.

- **Access** public

An function AnnotatedProperty::getAnnotations([\$name = null]) [line 90]

Function Parameters:

- **\$name \$name** The name of the annotation to filter on. Default is null.

Returns all property annotations. If a name is specified only annotations which match the specified name will be returned, otherwise all annotations are returned.

- **Access** public

AnnotatedClass function *AnnotatedProperty::getDeclaringClass()* [*line 137*]

Gets the parent class as an AnnotatedClass

- **Access** public

True function *AnnotatedProperty::hasAnnotation(\$annotation)* [*line 69*]

Function Parameters:

- *\$annotation* **\$annotation** The name of the annotation.

Checks the property for the presence of the specified annotation.

- **Access** public

True function *AnnotatedProperty::isAnnotated()* [*line 58*]

Returns boolean indicator based on the presence of any annotations.

- **Access** public

Class AnnotationParser

[line 32]

AgilePHP :: AnnotationParser Responsible for parsing and returning annotation details about PHP classes.

- **Package** com-makeabyte-agilephp-annotation
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc.
- **Static**

Array function AnnotationParser::getClassAnnotations(\$class) [line 163]

Function Parameters:

- [AnnotatedClass](#) \$class

Returns an array of class level annotations or false if no class level annotations are present.

- **Access** public

Array function AnnotationParser::getClassAnnotationsAsArray(\$class) [line 215]

Function Parameters:

- \$class \$class The name of the parsed class

Returns an array of class level annotations

- **Access** public

void function AnnotationParser::getInstance() [*line 46*]

- **Static**
- **Access** public

Array function AnnotationParser::getMethodAnnotations(\$method) [*line 196*]

Function Parameters:

- *\$method* **\$method** The AnnotatedMethod to search

Returns an array of method level annotations or false if no annotations are found for the specified method.

- **Access** public

Array function AnnotationParser::getMethodAnnotationsAsArray(\$class, 1) [*line 227*]

Function Parameters:

- *\$method* **1** The name of the parsed method
- *\$class* **\$class** The name of the parsed class

Returns an array of method level annotations for the specified class

- **Access** public

Array function AnnotationParser::getPropertyAnnotations(\$property) [*line 175*]

Function Parameters:

- *\$property* **\$property** The AnnotatedProperty to get the annotations for

Returns an array of property level annotations or false if no annotations are present for the specified property.

- **Access** public

Array function AnnotationParser::getPropertyAnnotationsAsArray(\$class, 1) [line 239]

Function Parameters:

- *\$property* **1** The name of the parsed property
- *\$class* **\$class** The name of the parsed class

Returns an array of property level annotations for the specified class

- **Access** public

PHP function AnnotationParser::getSourceCode() [line 436]

Returns the PHP file content to be parsed.

- **Access** public

void function AnnotationParser::parse(\$class) [line 61]

Function Parameters:

- *\$class* **\$class** The name of the class to parse

Breaks the class file into PHP tokens and extracts all interface, class, method, and property level annotations.

- **Access** public

Package com-makeabyte-agilephp-annotation-annotations Procedural Elements

AroundInvoke.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework
Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-annotation-annotations

Interceptor.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-annotation-annotations

RemoteMethod.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-annotation-annotations

Package com-makeabyte-agilephp-annotation-annotations Classes

Class AroundInvoke

[line 31]

AgilePHP :: AroundInvoke Annotation which causes #@Interceptor methods to be invoked

- **Package** com-makeabyte-agilephp-annotation-annotations
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Class Interceptor

[line 31]

AgilePHP :: Interceptor Annotation used to create an interceptor

- **Package** com-makeabyte-agilephp-annotation-annotations
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc.

Class RemoteMethod

[line 31]

AgilePHP :: RemoteMethod Annotation which authorizes a method for Remoting

- **Package** com-makeabyte-agilephp-annotation-annotations
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Package com-makeabyte-agilephp-identity

Procedural Elements

IdentityManager.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-identity

IdentityModel.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-identity

Role.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-identity

User.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-identity

Package com-makeabyte-agilephp-identity Classes

Class IdentityManager

[line 31]

AgilePHP :: IdentityManager Interface for Identity component

- **Package** com-makeabyte-agilephp-identity
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc.

void function IdentityManager::delete() [line 230]

Deletes the domain object model which the IdentityManager is managing.

- **Access** public

void function IdentityManager::forgotPassword() [line 146]

Sends the identity an email to the address stored in the stateful domain object model being managed by IdentityManager. Uses the state of the 'email' field within the domain object model the IdentityManager is managing.

- **Throws** AgilePHP_Exception If there was an error sending the forgotten password email.
- **Access** public

The function IdentityManager::getCreated() [line 107]

Returns the dateTime when the identity was created.

- **Access** public

The function IdentityManager::getEmail() [line 92]

Returns the identity's email address

- **Access** public

The function IdentityManager::getLastLogin() [line 122]

Returns the dateTime the identity last logged in

- **Access** public

An function IdentityManager::getModel() [line 46]

Returns the domain object model which the IdentityManager is managing.

- **Access** public

The function IdentityManager::getPassword() [line 77]

Returns the password for the identity which IdentityManager is managing. Unless overwritten, this is the persisted value.

- **Access** public

void function IdentityManager::getPasswordResetUrl() [*line 136*]

Returns the url which is sent to the identity that when clicked resets their password.

- **Access** public

The function IdentityManager::getRole() [*line 171*]

Returns the Role object

- **Access** public

The function IdentityManager::getUsername() [*line 61*]

Returns the username for the identity which IdentityManager is managing.

- **Access** public

True function IdentityManager::hasRole(\$role) [*line 179*]

Function Parameters:

- *\$role* **\$role** The name of a role

Checks to see if the identity has the specified role.

- **Access** public

True function IdentityManager::isLoggedIn() [*line 209*]

Returns boolean response based on the identity's logged in status.

- **Access** public

True function IdentityManager::login(\$username, \$password) [*line 195*]

Function Parameters:

- *\$username* **\$username** The username to authenticate
- *\$password* **\$password** The password to authenticate

Authenticates/logs in an identity and returns a boolean response.

- **Access** public

void function IdentityManager::logout() [*line 202*]

Destroys the current SessionScope.

- **Access** public

void function IdentityManager::merge() [*line 224*]

Updates the identity using the current state of the domain model object which the IdentityManager is managing.

- **Access** public

void function IdentityManager::persist() [line 216]

Persists a new identity.

- **Access** public

void function IdentityManager::refresh() [line 238]

Refreshes an entity by performing a 'find' operation on the domain object model which the IdentityManager is managing.

- **Access** public

void function IdentityManager::resetPassword(\$token, \$sessionId) [line 156]

Function Parameters:

- **\$token \$token** A randomly generated token required to reset the password
- **\$sessionId \$sessionId** The sessionId of the user who requested the new password

Resets the password to a hashed random string. This operation uses the AgilePHP Crypto component to ensure standard hashing across the application.

- **Access** public

void function IdentityManager::revokeRole() [line 186]
Revokes/removes a role from the identity.

- **Access** public

void function IdentityManager::setCreated(\$dateTime) [line 100]
Function Parameters:

- *\$dateTime* **\$dateTime** The dateTime when this identity was created.

Denotes when the identity was created.

- **Access** public

void function IdentityManager::setEmail(\$email) [line 85]
Function Parameters:

- *\$email* **\$email** A valid email addresss containing for the identity

Sets the identity's email address.

- **Access** public

void function IdentityManager::setLastLogin(\$dateTime) [line 115]
Function Parameters:

- *\$dateTime* **\$dateTime** The dateTime when the identity last logged in

Sets the dateTime when this identity last logged in

- **Access** public

void function IdentityManager::setModel(\$model) [*line 39*]

Function Parameters:

- *\$model* **\$model** The domain object model for IdentityManager to manage

Sets the domain object model which the IdentityManager will manage.

- **Access** public

void function IdentityManager::setPassword(\$password) [*line 69*]

Function Parameters:

- *\$password* **\$password** The identity's password

Sets the password for the identity which IdentityManager is managing.

- **Access** public

false function IdentityManager::setPasswordResetUrl(\$url) [*line 130*]

Function Parameters:

- **\$url \$url** The url which should reset the users password when its clicked

Sets the url used to reset the identity's password.

- **Access** public

void function IdentityManager::setRole(\$role) [line 164]

Function Parameters:

- **[Role](#) \$role** A Role domain model object

Sets the identity's role

- **Access** public

void function IdentityManager::setUsername(\$username) [line 54]

Function Parameters:

- **\$username \$username** The username of the identity.

Sets the username for the identity which IdentityManager is managing.

- **Access** public

Class IdentityModel

[line 31]

AgilePHP :: IdentityModel Interface for Identity domain model

- **Package** com-makeabyte-agilephp-identity
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

void function IdentityModel::getCreated() *[line 43]*

- **Access** public

void function IdentityModel::getEmail() *[line 40]*

- **Access** public

void function IdentityModel::getLastLogin() *[line 46]*

- **Access** public

void function IdentityModel::getPassword() *[line 37]*

- **Access** public

void function IdentityModel::getRole() [*line 49*]

- **Access** public

void function IdentityModel::getSessionId() [*line 52*]

- **Access** public

void function IdentityModel::getUsername() [*line 34*]

- **Access** public

void function IdentityModel::setCreated(\$dateTime) [*line 42*]

Function Parameters:

- **\$dateTime**

- **Access** public

void function IdentityModel::setEmail(\$email) [*line 39*]

Function Parameters:

- **\$email**

- **Access** public

void function IdentityModel::setLastLogin(\$dateTime) [line 45]

Function Parameters:

- **\$dateTime**
- **Access** public

void function IdentityModel::setPassword(\$password) [line 36]

Function Parameters:

- **\$password**
- **Access** public

void function IdentityModel::setRole(\$role) [line 48]

Function Parameters:

- [*Role*](#) **\$role**
- **Access** public

void function IdentityModel::setSessionId(\$sessionId) [line 51]

Function Parameters:

- **\$sessionId**

- **Access** public

void function IdentityModel::setUsername(\$username) [line 33]

Function Parameters:

- **\$username**

- **Access** public

Class Role

[line 31]

AgilePHP :: Role Role associated with an Identity

- **Package** com-makeabyte-agilephp-identity
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void function Role::__construct([\$role = null]) [line 36]*

Function Parameters:

- **\$role**

- **Access** public

The function Role::getDescription() [line 78]

Returns the role description

- **Access** public

The function Role::getName() [line 57]

Returns the name of the role

- **Access** public

void function Role::setDescription(\$description) [line 68]

Function Parameters:

- *\$description* **\$description** The role description

Sets the description of the role.

- **Access** public

void function Role::setName(\$name) [line 47]

Function Parameters:

- *\$name* **\$name** The role name

Sets the name of the role

- **Access** public

Class User

[line 31]

AgilePHP :: User User domain model

- **Package** com-makeabyte-agilephp-identity
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void* function User::User() *[line 45]*

- **Access** public

void function User::getCreated() *[line 82]*

- **Access** public

void function User::getEmail() *[line 72]*

- **Access** public

void function User::getLastLogin() [*line 92*]

- **Access** public

void function User::getPassword() [*line 62*]

- **Access** public

void function User::getRole() [*line 112*]

- **Access** public

void function User::getRoleId() [*line 102*]

- **Access** public

void function User::getRoles() [*line 122*]

- **Access** public

void function User::getSession() [*line 142*]

- **Access** public

void function User::getSessionId() [line 132]

- **Access public**

void function User::getUsername() [line 52]

- **Access public**

void function User::setCreated(\$dateTime) [line 77]

Function Parameters:

- **\$dateTime**

- **Access public**

void function User::setEmail(\$email) [line 67]

Function Parameters:

- **\$email**

- **Access public**

void function User::setLastLogin(\$timestamp) [line 87]

Function Parameters:

- **\$timestamp**

- **Access** public

void function User::setPassword(\$password) [*line 57*]

Function Parameters:

- **\$password**

- **Access** public

void function User::setRole(\$role) [*line 107*]

Function Parameters:

- [*Role*](#) **\$role**

- **Access** public

void function User::setRoleId(\$roleId) [*line 97*]

Function Parameters:

- **\$roleId**

- **Access** public

void function User::setRoles(\$roles) [line 117]

Function Parameters:

- **array \$roles**
- **Access public**

void function User::setSession(\$session) [line 137]

Function Parameters:

- [Session](#) **\$session**
- **Access public**

void function User::setSessionId(\$sessionId) [line 127]

Function Parameters:

- **\$sessionId**
- **Access public**

void function User::setUsername(\$username) [line 47]

Function Parameters:

- **\$username**

- **Access** public

Package com-makeabyte-agilephp-interception Procedural Elements

InterceptorProxy.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework
Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-interception

InvocationContext.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-interception

Package com-makeabyte-agilephp-interception Classes

Class InterceptorProxy

[line 31]

AgilePHP :: InterceptorProxy Dynamic proxy template for intercepted classes

- **Package** com-makeabyte-agilephp-interception
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc.

Constructor *void* function InterceptorProxy::__construct() [line 43]

Initializes the intercepted class by creating a new instance, passing in any constructor arguments as required. Class level interceptors are invoked here, as well as dependency injections via #@In.

- **Access** public

Singleton function InterceptorProxy::getInstance() [line 109]

Returns a singleton instance of the intercepted class

- **Static**
- **Access** public

The function `InterceptorProxy::getInterceptedInstance()` [line 125]

Returns the instance of the intercepted class.

- **Access** public

The function `InterceptorProxy::__call($method, $args)` [line 201]

Function Parameters:

- *\$method* **\$method** The method being called
- *\$args* **\$args** The arguments being passed

Magic PHP method executor.

- **Throws** `AgilePHP_InterceptionException`
- **Access** public

The function `InterceptorProxy::__get($property)` [line 137]

Function Parameters:

- *\$property* **\$property** The property/field name being gotten

Magic PHP property accessor

- **Throws** `AgilePHP_InterceptionException`
- **Access** public

True function InterceptorProxy::__isset(\$property) [line 176]

Function Parameters:

- *\$property* **\$property** The property/field being tested

Magic PHP isset. Checks a property to see if its set

- **Throws** AgilePHP_InterceptionException
- **Access** public

void function InterceptorProxy::__set(\$property, \$value) [line 157]

Function Parameters:

- *\$property* **\$property** The property/field name being set
- *\$value* **\$value** The value to set

Magic PHP property mutator

- **Throws** AgilePHP_InterceptionException
- **Access** public

void function InterceptorProxy::__unset(\$property) [line 188]

Function Parameters:

- *\$property* **\$property** The property/field being unset

Magic PHP unset function.

- **Throws** AgilePHP_InterceptionException
- **Access** public

Class InvocationContext

[line 31]

AgilePHP :: InvocationContext Provides environment and method execution details to interceptors.

- **Package** com-makeabyte-agilephp-interception
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

InvocationContext::\$proceed

mixed = false [line 38]

- **Access** public

Constructor *void* function InvocationContext::__construct(&\$target, \$method, \$parameters, &\$interceptor) [line 49]

Function Parameters:

- *\$target* &**\$target** The instance which was intercepted
- *\$method* **\$method** The name of the method which was intercepted
- *\$parameters* **\$parameters** The method parameters which were intercepted
- *\$interceptor* &**\$interceptor** The interceptor instance which caused the interception

Initializes the InvocationContext.

- **Access** public

Interceptor function `InvocationContext::getInterceptor()` [*line 118*]

Returns the instance of the interceptor annotation which caused the interception.

- **Access** public

The function `InvocationContext::getMethod()` [*line 62*]

Returns the name of the method which was intercepted.

- **Access** public

An function `InvocationContext::getParameters()` [*line 98*]

Returns the parameters being passed into the intercepted method.

- **Access** public

The function `InvocationContext::getTarget()` [*line 108*]

Returns the target instance which was intercepted.

- **Access** public

The function `InvocationContext::proceed()` [line 129]

Causes the intercepted invocation to proceed, using the method and parameters which are present in the current `InvocationContext`.

- **Access** public

void function `InvocationContext::setMethod($method)` [line 74]

Function Parameters:

- `$method` **\$method** The name of the method to invoke

Changes the name of the method. This is used by the interceptor to cause a different method to be executed than initially requested.

- **Access** public

void function `InvocationContext::setParameters($parameters)` [line 87]

Function Parameters:

- `array` **\$parameters** An array of parameters to pass into the invoked method

Changes the method parameters. This is used by the interceptor to cause a different set of parameters to be passed into the method which gets invoked.

- **Access** public

Package com-makeabyte-agilephp-interception-interceptors Procedural Elements

In.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework
Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-interception-interceptors

Restrict.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-interception-interceptors

Package com-makeabyte-agilephp-interception-interceptors Classes

Class In

[line 33]

AgilePHP :: In Dependency injection @In interceptor

- **Package** com-makeabyte-agilephp-interception-interceptors
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

In::\$class

mixed = [line 35]

- **Access** public

Class Restrict

[line 33]

AgilePHP :: Restrict Interceptor for @Restrict annotations

- **Package** com-makeabyte-agilephp-interception-interceptors
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Restrict::\$message

mixed = [line 37]

- **Access** public

Restrict::\$role

mixed = [line 35]

- **Access** public

Restrict::\$roles

mixed = [line 36]

- **Access** public

void function Restrict::requireRole(\$ic) [line 40]

Function Parameters:

- [InvocationContext](#) **\$ic**

- **Access** public

Package com-makeabyte-agilephp-mvc

Procedural Elements

AJAXRenderer.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework
Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-mvc

BaseController.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-mvc

BaseModelActionController.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-mvc

BaseModelController.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-mvc

BaseModelXmlController.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-mvc

BaseModelXslController.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-mvc

BaseRenderer.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-mvc

PHTMLRenderer.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-mvc

XSLTRenderer.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-mvc

Package com-makeabyte-agilephp-mvc

Classes

Class AJAXRenderer

[line 31]

AgilePHP :: MVC AJAXRenderer Renders data to JSON and XML with appropriate content-type header.

- **Package** com-makeabyte-agilephp-mvc
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc.

void function AJAXRenderer::render(\$data) [line 42]

Function Parameters:

- **\$data**

Renders the specified PHP data according to \$output. The appropriate content-type to the HTTP response header.
(non-PHPdoc)

- **See** render(\$view)
- **Access** public

void function AJAXRenderer::renderNoFormat(\$data) [line 107]

Function Parameters:

- **\$data \$data** The raw data to render

Renders the specified data without sending it through any of the AJAXRender internal formatting/converting methods. An appropriate HTTP content-type header is added to the response.

- **Access** public

void function AJAXRenderer::renderNoFormatNoHeader(\$data) [line 136]

Function Parameters:

- **\$data \$data** The data to render

Renders the specified data without sending it through any of the AJAXRenderer internal formatting/converting methods. In addition, no HTTP content-type header is added to the response.

- **Access** public

void function AJAXRenderer::renderNoHeader(\$data) [line 75]

Function Parameters:

- **\$data \$data** The data to render

Renders the specified data according to \$output without sending an HTTP content-

type header.

- **Access** public

void function AJAXRenderer::setOutput(\$type) [line 147]

Function Parameters:

- *\$type* **\$type** The data type output to the requestor. (XML|JSON)

Sets the desired output type.

- **Access** public

Class BaseController

[line 34]

AgilePHP :: MVC BaseController Provides basic renderer implementations and defines 'index' abstract method.

- **Package** com-makeabyte-agilephp-mvc
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc
- **Abstract Element**
- **Static**

Constructor *void function BaseController::__construct() [line 43]*

Creates a new instance of default renderer

- **Access** protected

void function BaseController::createCustomRenderer(\$renderer) [line 77]

Function Parameters:

- *\$renderer* **\$renderer** The custom renderer the controller will use to render views

Creates an instance of the specified custom renderer the controller will use to render views.

This renderer is loaded from the application 'classes' directory.

- **Access** protected

void function BaseController::createRenderer(\$renderer) [line 65]

Function Parameters:

- *\$renderer* **\$renderer** The framework renderer the controller will use to render views

Creates an instance of the specified renderer the controller will use to render views.

This renderer is loaded from the AgilePHP framework.

- **Access** protected

void function BaseController::getBaseJS([\$debug = false]) [line 89]

Function Parameters:

- **\$debug** **\$debug** True to enable client side AgilePHP debugging.

Returns the raw JavaScript contents of the AgilePHP.js file and pre-configures the library with a default AgilePHP.debug, AgilePHP.MVC.controller, and AgilePHP.MVC.action value.

- **Access** public

void function BaseController::getRenderer() [line 53]

Returns the controllers view renderer.

- **Access** protected

void function BaseController::index() [line 108]

Default controller action method.

- **Abstract Element**
- **Access** public

Class BaseModelActionController

[line 32]

AgilePHP :: MVC BaseModelActionController Provides a base implementation of a working model action controller.

- **Package** com-makeabyte-agilephp-mvc
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc
- **Abstract Element**

Constructor *void* function BaseModelActionController::__construct([\$requireLogon = true], [\$requiredRole = 'admin']) [*line 36*]

Function Parameters:

- **\$requireLogon**
- **\$requiredRole**

- **Access** protected

void function BaseModelActionController::delete() [*line 229*]

Deletes the current ActiveRecord defined by the model in the extension class.

- **Access** public

An function BaseModelActionController::getXsltRenderer() [*line 69*]

Returns an instance of the MVC XSLTRenderer object.

- **Access** protected

void function BaseModelActionController::merge() [*line 215*]

Merges the current ActiveRecord defined by the model in the extension class.

- **Access** public

void function BaseModelActionController::modelAction(\$action, [\$ids = ''], [\$page = 1], [\$view = 'admin'], 4) [line 87]

Function Parameters:

- *\$model* **4** A domain model object to invoke the action on. Use of this parameter requires the extension class to implement `DynamicModelController` found in the framework's core package.
- *\$action* **\$action** The action event type - add, edit, merge, delete
- *\$ids* **\$ids** The primary key values associated with the ActiveRecord.
- *\$page* **\$page** A page number to load (keeps result list state when performing updates/deletes)
- *\$view* **\$view** The 'view' to render. Default is 'admin'.

Handles the dispatching of action events from the view.

- **Access** public

void function BaseModelActionController::modelList([\$page = 1], [\$view = 'admin'], [\$model = null]) [line 158]

Function Parameters:

- *\$page* **\$page** The page number within the result set to display. Default is page 1.
- *\$view* **\$view** The view to render. Default is 'admin'.
- **\$model**

Performs a search for the model defined in the extension class and displays a paginated result list with edit and delete actions, as well as sortable column headers.

- **Access** public

void function BaseModelActionController::modelSort(\$column, \$direction, [\$page = 1], [\$view = 'admin']) [line 183]

Function Parameters:

- *\$column* **\$column** The column to sort
- *\$direction* **\$direction** The direction to sort. Default is 'ASC' (ascending)
- *\$page* **\$page** The page within the result set to display.
- *\$view* **\$view** The view to render. Default is 'admin'.

Sorts the modelList according to the specified column name and direction.

- **Access** public

void function BaseModelActionController::persist() [line 201]

Persists a new ActiveRecord defined by the model in the extension class.

- **Access** public

void function BaseModelActionController::revokeAccess([\$msg = 'Login required']) [line 59]

Function Parameters:

- *\$msg* **\$msg** The exception message. Default is 'Login required'.

Throws AgilePHP_NotLoggedInException

- **Throws** AgilePHP_NotLoggedInException
- **Access** protected

Class BaseModelController

[line 32]

AgilePHP :: MVC BaseModelController Provides base implementation for model controllers.

- **Package** com-makeabyte-agilephp-mvc
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc
- **Abstract Element**

Constructor *void* function BaseModelController::__construct() [line 39]

Initializes parent and executes a default SELECT query for the first 25 records.

- **Access** protected

void function BaseModelController::createQuery(\$sql) [line 109]

Function Parameters:

- ~~\$sql~~ **\$sql** A valid SQL statement

Sets the SQL statement to use when calling executeQuery.

- **Access** protected

void function BaseModelController::delete() [line 357]

Deletes the model ActiveRecord defined in the extension class. The model must have its primary key properties defined for this operation to succeed.

- **Access** protected

void function BaseModelController::executeCountQuery() [line 275]

Executes an SQL count query for total number of records in the database.
Initializes 'pageCount' and 'count' properties.

- **Access** protected

void function BaseModelController::executeQuery() [line 285]

Executes the current SQL statement as defined in 'sql'.

- **Access** protected

A function BaseModelController::find([\$model = null]) [line 324]

Function Parameters:

- **\$model**

Search for an ActiveRecord for the model defined in the extension class.

- **Access** protected

Total function BaseModelController::getCount() [*line 201*]

Returns the total number of records in the database table

- **Access** protected

The function BaseModelController::getMaxResults() [*line 99*]

Returns the maximum number of results to retrieve. This translates to an SQL LIMIT clause during SELECT operations.

- **Access** protected

void function BaseModelController::getModel() [*line 468*]

- **Abstract Element**
- **Access** protected

The function BaseModelController::getModelName() [*line 306*]

Returns the model's class name.

- **Access** protected

void function BaseModelController::getNextResultList() [*line 254*]

Sets the result list up with a 'next page' of a pagination request

- **Access** protected

An function BaseModelController::getOrderBy() [line 191]

Returns an associative array containing the current 'orderBy' clause. The results are returned with the name of the column as the index and the direction as the value.

- **Access** protected

The function BaseModelController::getPage() [line 211]

Returns the current page number

- **Access** protected

The function BaseModelController::getPageCount() [line 222]

Returns the total number of pages. This is calculated by dividing the total number of records by 'maxResults'.

- **Access** protected

PersistenceManager function BaseModelController::getPersistenceManager() [line 56]

Returns an instance of PersistenceManager

- **Throws** AgilePHP_Exception if the instance of PersistenceManager is null
- **Access** protected

void function BaseModelController::getPreviousResultList() [line 264]

Sets the result list up with a 'previous page' of a pagination request

- **Access** protected

The function BaseModelController::getQuery() [line 119]

Returns the current SQL query

- **Access** protected

The function BaseModelController::getResultCount() [line 66]

Returns the total number of records in the current result list.

- **Access** protected

An function BaseModelController::getResultList() [line 77]

Returns a result set from the database. The result list is returned as an array of stdClass objects, each representing a row in the database.

- **Access** protected

Result function BaseModelController::getResultListAsModels([\$populateForeignModels = false]) [line 398]

Function Parameters:

- *\$populateForeignModels* **\$populateForeignModels** If set to true, a lookup is performed on foreign models.

Returns the result list as an array of populated model objects. If the model contains any foreign keys, by default the foreign model has only its primary keys which exist as foreign keys in the current model populated and the foreign model is added to its instance property in the current ActiveRecord.

- **Access** protected

The function BaseModelController::getTableName() [line 295]

Returns the table name for the model defined in the extension class

- **Access** protected

void function BaseModelController::merge() [line 346]

Merges the model defined in the extension class. The model must have its primary key properties defined for this operation to succeed.

- **Access** protected

True function BaseModelController::nextExists() [line 233]

Returns boolean result based on whether or not a 'next page' result is available during a pagination request.

- **Access** protected

void function BaseModelController::persist() [line 335]

Persists a new model ActiveRecord defined in the extension class.

- **Access** protected

True function BaseModelController::previousExists() [line 244]

Returns boolean result based on whether or not a 'previous page' result is available during a pagination request.

- **Access** protected

void function BaseModelController::setGroupBy(\$column) [line 167]

Function Parameters:

- *\$column* **\$column** The column name to group the result set by

Sets the SQL 'group by' clause.

- **Access** protected

void function BaseModelController::setMaxResults(\$count) [line 88]

Function Parameters:

- *\$count* **\$count** Maximum number of records to return

Sets the maximum number of records to return in a result list

- **Access** protected

void function BaseModelController::setOrderBy(\$column, \$direction) [line 179]

Function Parameters:

- *\$column* **\$column** The column name to order the result set by
- *\$param* **\$direction** The direction to sort the result set (ASC|DESC).
- **\$direction**

Sets the SQL 'order by' clause.

- **Access** protected

void function BaseModelController::setPage([\$pageNumber = 1]) [line 131]

Function Parameters:

- *\$pageNumber* **\$pageNumber** The page number. Default is 1.

Sets the pagination page number and performs an SQL query to populate the 'resultList' and 'resultCount' properties with their appropriate values for the specified page.

- **Access** protected

void function BaseModelController::setRestrictions(\$restrictions) [line 145]

Function Parameters:

- **\$restrictions** **\$restrictions** An associative array containing WHERE clause restrictions. (For example: array('id' => 21))

Sets WHERE clause restrictions

- **Access** protected

void function BaseModelController::setRestrictionsLogicOperator(\$operator) [*line 156*]

Function Parameters:

- **\$operator** **\$operator** The logical operator 'and'/'or' to be used in SQL WHERE clause. Default is 'AND'.

Sets the restriction operator (and|or) used in SQL WHERE clause.

- **Access** protected

The function BaseModelController::toAccessor(\$property) [*line 370*]

Function Parameters:

- **\$property** **\$property** The name of the property to convert to an accessor method name

Creates an accessor method from the \$property parameter. The \$property will be returned with the prefix 'get' and the first letter of the property upcased.

- **Access** protected

The function BaseModelController::toMutator(\$property) [line 383]

Function Parameters:

- **\$property \$property** The name of the property to convert to a mutator method name

Creates a mutator method from the \$property parameter. The \$property will be returned with the prefix 'set' and the first letter of the property upcased.

- **Access** protected

Class BaseModelXmlController

[line 32]

AgilePHP :: MVC BaseModelXmlController Provides base implementation for model xml controllers.

- **Package** com-makeabyte-agilephp-mvc
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc
- **Abstract Element**

Constructor *void* function BaseModelXmlController::__construct() [line 34]

- **Access** protected

void function BaseModelXmlController::getModelAsFormXML([\$controller = null], [\$action = null], [\$params = null])
[line 54]

Function Parameters:

- **\$controller \$controller** The controller to use for add/update/delete operations. Defaults to the controller that invoked this method.
- **\$action \$action** The controllers action method to invoke. Defaults to the model name (lowercase) followed by the action mode 'Add' or 'Edit' which is determined by whether or not an 'id' parameter has been specified. For example, a user model would be either 'userAdd' or 'userEdit' depending on whether or not an 'id' parameter is specified. If no 'id' has been specified the default action would be 'userAdd', or 'userEdit' if an 'id' parameter has been set.
- **\$params \$params** An array of parameters to

Creates an XML document representing a model. If the 'id' parameter is set, a lookup is performed for the model with the specified 'id' and the XML is returned with data populated from the database result set.. If there is no 'id' set, the model's property nodes will be null. A custom controller and action can be set to modify default behavior.

- **Access protected**

'persist' function BaseModelXmlController::getModelPersistenceAction() [line 328]

Returns the type of action which the controllers should take when deciphering whether the operation is a persist or merge operation. If the primary key(s) contain a value, the action is assumed a merge. If the primary key(s) do not contain a value, the action is assumed persist.

- **Access protected**

An function BaseModelXmlController::getResultListAsPagedXML([\$controller = null], [\$action = null], [\$params = null]) [line 190]

Function Parameters:

- **\$controller**
- **\$action**
- **\$params**

Returns a paged result set from the database as XML (including foreign model instances with their primary keys set). The XML document is returned in the following format:

```

    <ResultList>      <Model>          <your_model_name>
<foreign_model_name>    <primary_key1>*</primary_key1>
<primary_key2>*</primary_key2>    </foreign_model_name>
<model_prop1>*</model_prop1>    <model_prop2>*</model_prop2>
    </your_model_name>          </Model>          <Pagination>
<page>*</page>          <pageCount>*</pageCount>
<nextExists>*</nextExists>    <previousExists>*</previousExists>
    <controller>*</controller>          <action>*</action>
</Pagination> </ResultList>

```

- Access protected

An function BaseModelXmlController::getResultListAsXML() [*line 132*]

Returns a result set from the database as XML. The XML document is returned with the root node 'ResultList' containing an element named after the model which then contains each of the models properties and values as children.

For example: <ResultList> <your_model_name>
 <model_prop1>*</model_prop1> <model_prop2>*</model_prop2>
 </your_model_name> </ResultList>.

- Access protected

Class BaseModelXslController

[*line 32*]

AgilePHP :: MVC BaseModelXslController Provides base implementation for model xsl controllers.

- **Package** com-makeabyte-agilephp-mvc
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc
- **Abstract Element**

Constructor *void* function BaseModelXslController::__construct() [*line 34*]

- **Access** protected

An function BaseModelXslController::getModelAsReadOnlyXSL() [*line 256*]

Returns an XSL stylesheet used for read-only (this is the Read in CRUD).

- **Access** protected

void function BaseModelXslController::getModelFormXSL(0) [*line 230*]

Function Parameters:

- *\$pkeyValues* **0** The primary key value(s) associated with this record

Returns an XSL stylesheet used for add and update actions. During an 'add' operation

\$pkeyValues should be null. During an 'update' action, *\$pkeyValues* should be defined with the primary key value associated with the record. If the record contains a compound primary key, *\$pkeyValues* should be defined as an associative array containing each of the primary key names with their corresponding values (ie. array('key1' => 'val1', ...))

- **Access** protected

XSL function BaseModelXslController::getModelListXSL([\$keyFields = "], [\$controller = "], [\$action = "]) *[line 52]*

Function Parameters:

- *\$keyFields* **\$keyFields** The name of the model property to send as the 'id' field to the action when an action button is clicked. Defaults to the primary key(s) of the model as defined in persistence.xml.
- *\$controller* **\$controller** The name of the controller to use when an action button is clicked. Defaults to the name of the controller which invoked this method.
- *\$action* **\$action** The name of a custom action to invoke within the controller when an action button is clicked. Defaults to the name of the model (lowercase) followed by the string 'Action' (ie, 'User' model would be 'userAction').

Generates an XSL stylesheet from the domain model object's properties. Designed to be used in conjunction with getResultListAsPagedXML() to perform an XSLT transformation.

- **Access** protected

Pagination function BaseModelXslController::getPaginationXSL() *[line 313]*

Returns an XSL stylesheet used for pagination.

- **Access** protected

Class BaseRenderer

[line 32]

AgilePHP :: MVC BaseRenderer Provides base rendering implementation

- **Package** com-makeabyte-agilephp-mvc
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc
- **Abstract Element**

The function BaseRenderer::get(\$key) [line 55]

Function Parameters:

- **\$key \$key** The key to retrieve the value from

Returns the value for the specified key.

- **Access** public

void function BaseRenderer::getStore() [line 66]

Returns the store which contains variable names with their associated values set by one or more controllers.

- **Access** public

void function BaseRenderer::render(\$view) [line 71]

Function Parameters:

- **\$view**

- **Abstract Element**
- **Access** public

void function BaseRenderer::set(\$key, \$value) [line 44]

Function Parameters:

- **\$key \$key** The variable name
- **\$value \$value** The variable value

Helper method to allow controllers to set variables which are dumped to a view during rendering.

- **Access** public

Class PHTMLRenderer

[line 31]

AgilePHP :: MVC PHTMLRenderer Provides base rendering implementation for PHTML (PHP) views.

- **Package** com-makeabyte-agilephp-mvc
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

void function PHTMLRenderer::render(\$view) [line 40]

Function Parameters:

- **\$view \$view** The view which is rendered from the web app's 'view' directory

Renders a view by dumping all 'store' variables to locally scoped (page) variables. The view is expected to be in <webapp>/view.

- **Access** public

unknown_type function PHTMLRenderer::renderComponent(\$componentName, \$view) [*line 68*]

Function Parameters:

- *\$componentName* **\$componentName**
- **\$view**

Renders a component view by dumping all 'store' variables to locally scoped (page) variables. The view is expected to be in <webapp>/component/<component_name>/view.

- **Access** public

Class XSLTRenderer

[*line 31*]

AgilePHP :: MVC XSLTRenderer Provides base implementation for XSLT transformations

- **Package** com-makeabyte-agilephp-mvc
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

void function XSLTRenderer::ErrorHandler(\$errno, \$errmsg, \$errfile, \$errline) [line 157]

Function Parameters:

- **\$errno**
- **\$errmsg**
- **\$errfile**
- **\$errline**

loadXml reports an error instead of throwing an exception when the xml is not well formed. This is a custom PHP error handling function which throws an AgilePHP_Exception instead of reporting a PHP error.

- **Static**
- **Access** public

void function XSLTRenderer::getRenderedXslFile(\$xsl, [\$xml = ""]) [line 133]

Function Parameters:

- **\$xsl/ \$xsl** XSL document located in the web app view directory
- **\$xml/ \$xml** XML document as a string type

Performs an XSLT tranformation using a specified XSL view document and returns the HTML result.

The XSL document is read in from the web application 'view' directory. The XML document is passed in as a string.

- **Access** public

void function XSLTRenderer::render(\$xsl, [\$xml = ""]) [line 41]

Function Parameters:

- **\$xsl**/**\$xsl** XSL document as a string type
- **\$xml**/**\$xml** XML document as a string type

Renders a view by performing an XSLT transformation. The XSL document is passed in as a string. The XML document is passed in as a string.

- **Access** public

void function XSLTRenderer::renderXslFile(\$xsl, [\$xml = ""]) [line 105]

Function Parameters:

- **\$xsl**/**\$xsl** XSL document located in the web app view directory
- **\$xml**/**\$xml** XML document as a string type

Renders a view by performing an XSLT transformation. The XSL document is read in from the specified xsl view. The XML document is passed in as a string.

- **Access** public

The function XSLTRenderer::transform(\$xsl, [\$xml = ""]) [line 68]

Function Parameters:

- **\$xsl**/**\$xsl** XSL document as a string type
- **\$xml**/**\$xml** XML document as a string type

Performs an XSLT transformation and returns the rendered HTML.

- **Access** public

Package com-makeabyte-agilephp-persistence Procedural Elements

BasePersistence.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework
Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-persistence

require_once ['ForeignKey.php'](#)*[line 29]*

require_once ['Column.php'](#)*[line 28]*

require_once ['Table.php'](#)*[line 27]*

require_once ['Database.php'](#)*[line 26]*

require_once ['PersistenceRenderer.php'](#)*[line 25]*

Includes all persistence package dependancies

Column.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-persistence

Database.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-persistence

ForeignKey.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-persistence

PersistenceRenderer.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-persistence

Table.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-persistence

Package com-makeabyte-agilephp-persistence Classes

Class BasePersistence

[line 41]

AgilePHP :: BasePersistence Base persistence class responsible for all database/ORM actions

- **Package** com-makeabyte-agilephp-persistence
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc
- **Abstract Element**

BasePersistence::\$database

mixed = [line 48]

- **Access** protected

BasePersistence::\$model

mixed = [line 47]

- **Access** protected

BasePersistence::\$pdo

mixed = [line 46]

- **Access** protected

BasePersistence::\$transactionInProgress

mixed = [line 49]

- **Access** protected

Constructor *void* function BasePersistence::__construct() [line 51]

- **Access** protected

void function BasePersistence::beginTransaction() [line 136]

Begins a transaction

- **See** <http://usphp.com/manual/en/function.pdo-begintransaction.php>
- **See** <http://us2.php.net/manual/en/pdo.transactions.php>
- **Throws** AgilePHP_PersistenceException
- **Access** public

void function BasePersistence::close() [line 645]

Closes the connection to the database.

- **Access** public

void function BasePersistence::commit() [*line 158*]

Commits an already started transaction.

- **See** <http://usphp.com/manual/en/function.pdo-commit.php>
- **See** <http://us2.php.net/manual/en/pdo.transactions.php>
- **Throws** AgilePHP_PersistenceException
- **Access** public

True function BasePersistence::compare(\$a, \$b) [*line 946*]

Function Parameters:

- **\$a \$a** The first object
- **\$b \$b** The second object

Compares domain model object \$a with \$b.

NOTE: This function assumes the model adheres to the property/getter/setter model convention.

- **Access** public

PDOStatement function BasePersistence::delete(\$model) [*line 461*]

Function Parameters:

- **\$model \$model** The domain model object to delete

Deletes a persisted domain model object (ActiveRecord)

- **Throws** AgilePHP_PersistenceException
- **Access** public

The function BasePersistence::exec(\$statement) [line 273]

Function Parameters:

- *\$statement* **\$statement** The SQL statement to execute.

Executes an SQL statement and returns the number of rows affected by the query.

- **See** <http://usphp.com/manual/en/function.PDO-exec.php>
- **Access** public

True function BasePersistence::execute([\$inputParameters = array()]) [line 237]

Function Parameters:

- *\$inputParameters* **\$inputParameters** An array of input parameters

Executes a prepared statement (with parameters)

- **See** <http://usphp.com/manual/en/function.PDOStatement-execute.php>
- **Access** public

void function BasePersistence::find(\$model, [\$findAll = false]) [line 509]

Function Parameters:

- *\$model* **\$model** A domain model object with its primary key field(s) set

- **\$findAll**

Attempts to locate the specified model by primary key value.

- **Throws** AgilePHP_PersistenceException If any primary keys contain null values or any errors are encountered executing queries
- **Access** public

void function BasePersistence::getAvailableDrivers() [line 68]

Returns a list of drivers available to PDO.

- **Access** public

The function BasePersistence::getColumnnameForProperty(\$table, \$propertyName) [line 871]

Function Parameters:

- *\$table* **\$table** The 'Table' object containing the 'Column' to search.
- *\$propertyName* **\$propertyName** The name of the property to retrieve the name attribute value from

Returns the value of the 'name' attribute configured in persistence.xml for the specified \$propertyName.

If the property attribute does not exist, a match is attempted against the column name. If the column name matches the expected \$propertyName, the column name is returned. If neither can be matched, null is returned instead.

- **Access** public

The function `BasePersistence::getDatabase()` [line 78]

Returns the 'Database' object being used by the persistence framework.

- **Access** public

Custom function `BasePersistence::getDisplay_name_for_column($table, $columnName)` [line 820]

Function Parameters:

- *\$table* **\$table** The 'Table' object which contains the column to retrieve the display name from.
- *\$columnName* **\$columnName** The name of the column to get the display name for

Returns a custom display name as configured in persistence.xml 'display' attribute for the specified column name. If the 'Column' name can not be matched, it is then compared against the 'Column' 'property' attribute value. If neither can be matched, the \$columnName is returned.

- **Access** public

The function `BasePersistence::getDistinct()` [line 100]

Returns the 'distinct' column to use in an SQL SELECT statement if one has been defined.

- **Access** public

An function `BasePersistence::getIdentityModel()` [line 733]

Returns an instance of the domain model object responsible for AgilePHP 'Identity' persistence.

- **Access** public

The function BasePersistence::getIdentityTable() [line 715]

Returns a 'Table' object representing the table configured in persistence.xml as the AgilePHP 'Identity' table.

- **Access** public

The function BasePersistence::getMaxResults() [line 123]

Returns the 'maxResults' property value which is used during a 'find' operation which contains an empty model.

- **Access** public

The function BasePersistence::getPDOInstance() [line 58]

Returns the PDO instance in use by the persistence framework.

- **Access** public

The function BasePersistence::getPropertyNameForColumn(\$table, \$columnName) [line 848]

Function Parameters:

- **\$table \$table** The 'Table' object containing the 'Column' to search.
- **\$columnName \$columnName** The name of the column to retrieve the property attribute value from

Returns the value of the 'property' attribute configured in persistence.xml for the

specified \$columnName.

If the property attribute does not exist, the column name is returned instead.

- **Access** public

An function BasePersistence::getSessionModel() [line 795]

Returns an instance of the domain model object responsible for AgilePHP 'SessionScope' session persistence.

- **Access** public

The function BasePersistence::getSessionTable() [line 777]

Returns the 'Table' object that represents the table configured in persistence.xml as an AgilePHP 'SessionScope' session table.

- **Access** public

The function BasePersistence::getTableByModel([\$model = null]) [line 658]

Function Parameters:

- **\$model \$model** The domain model object to retrieve the table element for. Defaults to the model currently being managed by the 'PersistenceManager'.

Returns the 'Table' object which is mapped to the specified 'Model'.

- **Access** public

The function BasePersistence::getTableByModelName(\$modelName) [line 684]

Function Parameters:

- *\$modelName* **\$modelName** The name of the model

Returns the 'Table' object responsible for the specified \$modelName.

- **Access** public

The function BasePersistence::getTableByName(\$tableName) [line 699]

Function Parameters:

- *\$tableName* **\$tableName** The value of the table's 'name' attribute

Returns a 'Table' object by its name as configured in persistence.xml

- **Access** public

The function BasePersistence::isEmpty(\$model) [line 921]

Function Parameters:

- **\$model**

Checks the model in use by the persistence framework for the presence of property values. If the model does not contain any values, it is considered empty.

- **Access** public

PDOStatement function BasePersistence::merge(\$model) [*line 398*]

Function Parameters:

- *\$model* **\$model** The model object to merge/update

Merges/updates a persisted domain model object

- **Throws** AgilePHP_PersistenceException
- **Access** public

PDOStatement function BasePersistence::persist(\$model) [*line 330*]

Function Parameters:

- *\$model* **\$model** The domain model object to persist

Persists a domain model object

- **Throws** AgilePHP_PersistenceException
- **Access** public

False function BasePersistence::prepare(\$statement) [*line 205*]

Function Parameters:

- *\$statement* **\$statement** The SQL statement to prepare

Prepares an SQL prepared statement

- See <http://usphp.com/manual/en/function.pdo-prepare.php>
- Access public

The function BasePersistence::query(\$sql) [line 290]

Function Parameters:

- \$sql **\$sql** The SQL statement to execute

Executes a raw SQL query

- See <http://usphp.com/manual/en/function.pdo-query.php>
- Throws AgilePHP_PersistenceException
- Access public

The function BasePersistence::quote(\$data) [line 318]

Function Parameters:

- \$data **\$data** The data to quote

Quotes a string so its theoretically safe to pass into a statement

- See <http://www.php.net/manual/en/pdo.quote.php>
- Access public

void function BasePersistence::rollBack([\$message = null], [\$code = 0]) [line 182]

Function Parameters:

- *\$message* **\$message** Error/reason why the transaction was rolled back
- *\$code* **\$code** An error/reason code

Rolls back a transaction.

- **See** <http://usphp.com/manual/en/function.PDO-rollBack.php>
- **See** <http://us2.php.net/manual/en/pdo.transactions.php>
- **Throws** `AgilePHP_PersistenceException`
- **Access** public

void function BasePersistence::setDistinct(\$columnName) [line 89]

Function Parameters:

- *\$columnName* **\$columnName** The column name to get the distinct values for

Adds an SQL distinct clause to 'find' operation.

- **Access** public

void function BasePersistence::setMaxResults([\$maxResults = 25]) [line 112]

Function Parameters:

- *\$maxResults* **\$maxResults** The maximum number of results to return

Sets the 'maxResults' property value which is used during a 'find' operation which contains an empty model.

- **Access** public

The function BasePersistence::toAccessor(\$property) [line 988]

Function Parameters:

- *\$property* **\$property** The name of the property to convert to an accessor method name

Creates an accessor method from the \$property parameter. The \$property will be returned with the prefix 'get' and the first letter of the property uppercased.

- **Access** public

bigint function BasePersistence::toBigInt(\$number) [line 894]

Function Parameters:

- *\$int* **\$number** The bigint value

Converts the specified parameter to a bigint.

- **Access** public

The function BasePersistence::toMutator(\$property) [line 1001]

Function Parameters:

- *\$property* **\$property** The name of the property to convert to a mutator method name

Creates a mutator method from the \$property parameter. The \$property will be returned with the prefix 'set' and the first letter of the property uppercased.

- **Access** public

PDOStatement function BasePersistence::truncate(\$model) [*line 494*]

Function Parameters:

- *\$model* **\$model** A domain model object

Truncates the table for the specified domain model object

- **Throws** AgilePHP_PersistenceException
- **Access** public

void function BasePersistence::__destruct() [*line 909*]

Closes the connection to the database.

- **Access** public

Class Column

[*line 31*]

AgilePHP :: Column Represents a database table column in the AgilePHP persistence component.

- **Package** com-makeabyte-agilephp-persistence
- **Author** Jeremy Hahn

- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void* function Column::__construct(\$column, \$tableName) [*line 49*]

Function Parameters:

- *SimpleXMLElement* **\$column**
- **\$tableName**

- **Access** public

void function Column::getDefault() [*line 216*]

Returns the default value assigned to the column data when a value has not been specified.

- **Access** public

void function Column::getDescription() [*line 151*]

Returns the friendly description of the column.

- **Access** public

The function Column::getDisplay() [*line 194*]

Returns the friendly display name for the column

- **Access** public

True function Column::getForeignKey() [*line 394*]

Returns boolean indicator based on whether or not the column is a foreign key field.

- **Access** public

The function Column::getLength() [*line 129*]

Returns the length of the data being stored in the column

- **Access** public

Property function Column::getModelPropertyName() [*line 420*]

Returns the name which is used to access/mutate model properties/fields. If a property attribute has been configured in persistence.xml for the column, the property value is returned, otherwise the column name is returned instead.

- **Access** public

void function Column::getName() [*line 85*]

Returns the name of the column in the physical database

- **Access** public

The function Column::getProperty() [*line 172*]

Returns the name of the property in the domain model which the column maps

- **Access** public

The function Column::getType() [line 108]

Returns the data type which describes the type of data which is being stored in the column/field.

- **Access** public

The function Column::getViewDisplayName() [line 432]

Returns the value which is returned to the presentation tier when displaying the name of the column. If persistence.xml contains a valid 'display' attribute, this value is returned, otherwise, the column name is returned.

- **Access** public

True function Column::hasForeignKey() [line 384]

Returns boolean indicator based on whether or not the column is a foreign key field.

- **Access** public

True function Column::isAutoIncrement() [line 353]

Returns boolean flag indicating whether or not this column is an AUTO_INCREMENT field.

- **Access** public

void function Column::isBit() [line 405]

Returns boolean flag indicating whether or not the column is of the data type 'bit'. This is a **special data type which is used to render HTML checkboxes by the presentation tier.**

- **Access** public

True function Column::isForeignKey() [line 374]

Returns boolean indicator based on whether or not the column is a foreign key field.

- **Access** public

True function Column::isPrimaryKey() [line 331]

Returns boolean flag indicating whether or not the column is a primary key

- **Access** public

void function Column::isRequired() [line 310]

Returns boolean flag indicating whether or not the column data is required

- **Access** public

True function Column::isSelectable() [*line 289*]

Returns boolean flag indicating whether or not the column can be used by child tables to render data in an HTML drop-down instead of their foreign key value(s)

- **Access** public

True function Column::isSortable() [*line 263*]

Returns boolean flag indicating whether or not the column is sortable.

- **Access** public

True function Column::isVisible() [*line 240*]

Returns boolean indicator based on the visibility of the column.

- **Access** public

void function Column::setAutoIncrement(\$boolean) [*line 342*]

Function Parameters:

- *\$boolean* **\$boolean** True if the column contains AUTO_INCREMENT values, false otherwise

Marks the column as an AUTO_INCREMENT column.

- **Access** public

void function Column::setDefault(\$value) [line 205]

Function Parameters:

- *\$value* **\$value** The default value if nothing has been assigned

Sets the default value of the column data

- **Access** public

void function Column::setDescription(\$description) [line 141]

Function Parameters:

- *\$description* **\$description** The friendly description explaining the purpose of the column, etc.

Sets the friendly description of the column. This is used by visual elements to describe the column to the end user or website owner.

- **Access** public

void function Column::setDisplay(\$text) [line 184]

Function Parameters:

- *\$text* **\$text** The friendly name to display

Sets the friendly display name of the column. This is used by visual elements (such as forms) to show a user friendly name for the column.

- **Access** public

void function Column::setForeignKey(\$foreignKey) [line 364]

Function Parameters:

- *\$foreignKey* **\$foreignKey** True to mark the column as a foreign key column, false otherwise

Marks the column as a foreign key column.

- **Access** public

void function Column::setLength(\$length) [line 119]

Function Parameters:

- *\$length* **\$length** The maximum length of the data being persisted

Sets the length of the data being stored in the column

- **Access** public

void function Column::setName(\$name) [line 75]

Function Parameters:

- *\$name* **\$name** The column name in the physical database

Sets the name of the column in the physical database

- **Access** public

void function Column::setPrimaryKey(\$boolean) [line 321]

Function Parameters:

- *\$boolean* **\$boolean** True to mark this column as a primary key, false otherwise

Marks this column as a primary key

- **Access** public

void function Column::setProperty(\$property) [line 162]

Function Parameters:

- *\$property* **\$property** The property name in the domain model that stores the column data

Sets the name of the property in the domain model which the column maps

- **Access** public

void function Column::setRequired(\$boolean) [line 300]

Function Parameters:

- *\$boolean* **\$boolean** True if this column requires a value, false otherwise

Boolean flag indicating whether or not the column data is required

- **Access** public

void function Column::setSelectable(\$boolean) [line 277]

Function Parameters:

- *\$boolean* **\$boolean** Allows child tables to use the values in this column in an HTML drop-down in place of their foreign key value(s)

When set to true, this option allows tables who are a child table referencing this column, to render this columns data instead of its foreign key values.

- **Access** public

void function Column::setSortable(\$boolean) [line 252]

Function Parameters:

- *\$boolean* **\$boolean** True to enable sortable column header, false to render column header as plain text.

Enables/disables sortable column headers,

- **Access** public

void function Column::setType(\$type) [line 97]

Function Parameters:

- *\$type* **\$type** The data type being stored (varchar|int|text|etc...)

Sets the data type which describes the type of data which is to be stored in the column/field.

- **Access** public

void function Column::setVisible(\$boolean) [line 230]

Function Parameters:

- *\$boolean* **\$boolean** The boolean flag indicating whether or not the column should be rendered. True to render, false to hide.

Sets a boolean flag used to toggle the visibility of the column. This is used to hide a particular field (such as a primary key) from a form when its rendered.

- **Access** public

Class Database

[line 31]

AgilePHP :: Database Represents a database in the AgilePHP persistence component.

- **Package** com-makeabyte-agilephp-persistence
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void* function Database::__construct([\$database = null]) [*line 42*]

Function Parameters:

- *SimpleXMLElement* **\$database**

- **Access** public

void function Database::addTable(\$table) [*line 205*]

Function Parameters:

- [*Table*](#) **\$table** The table instance to push onto the stack

Pushes a new Table instance onto the stack

- **Access** public

The function Database::getHostname() [*line 139*]

Returns the hostname of the database server.

- **Access** public

void function Database::getId() [*line 75*]

Returns the unique database identifier.

- **Access** public

The function Database::getName() [line 96]

Returns the name of the database

- **Access** public

The function Database::getPassword() [line 181]

Returns the password thats used to connect to the database

- **Access** public

Array function Database::getTables() [line 227]

Returns an array of Table instances which represent a table in the physical database.

- **Access** public

The function Database::getType() [line 117]

Returns the database server type.

- **Access** public

The function Database::getUsername() [line 160]

Returns the username which is used to connect to the database

- **Access** public

True function Database::hasTables() [*line 216*]

Returns boolean indicator based on the presence of any Table instances configured for the database.

- **Access** public

void function Database::setHostname(\$hostname) [*line 129*]

Function Parameters:

- *\$hostname* **\$hostname** The hostname of the database server (or file path for sqlite databases).

Sets the hostname of the database server

- **Access** public

void function Database::setId(\$id) [*line 65*]

Function Parameters:

- *\$id* **\$id** The database identifier. This can be any legal XML value.

Sets the database id. This is used internally by AgilePHP to distinguish between multiple data sources configured in persistence.xml

- **Access** public

void function Database::setName(\$name) [line 86]

Function Parameters:

- *\$name* **\$name** The database name

Sets the name of the database

- **Access** public

void function Database::setPassword(\$password) [line 171]

Function Parameters:

- *\$password* **\$password** The password used to authenticate access to the database

Sets the password which is used to connect to the database

- **Access** public

void function Database::setTables(\$tables) [line 194]

Function Parameters:

- *array* **\$tables** An array of Table instances which represent a table in the physical database.

Sets the array of Table instances which represent a table in the physical database.

- **Access** public

void function Database::setType(\$type) [line 107]

Function Parameters:

- *\$type* **\$type** The database server type (sqlite|mysql|pgsql|firebird|informix|oracle|dblib|ibm)

Sets the type of database server

- **Access** public

void function Database::setUsername(\$username) [line 150]

Function Parameters:

- *\$username* **\$username** The username to connect to the database with

Sets the username required to access the database

- **Access** public

Class ForeignKey *[line 31]*

AgilePHP :: ForeignKey Represents a foreign key in the AgilePHP persistence component.

- **Package** com-makeabyte-agilephp-persistence
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void* function ForeignKey::__construct(\$foreign, \$tableName, \$columnName) [*line 46*]

Function Parameters:

- *SimpleXMLElement* **\$foreign**
- **\$tableName**
- **\$columnName**

- **Access** public

The function ForeignKey::getColumnInstance() [*line 317*]

Returns the column instance where the foreign key resides

- **Access** public

The function ForeignKey::getFkColumn() [*line 254*]

Returns the physical name of the column which stores the foreign key.

- **Access** public

The function ForeignKey::getFkTable() [line 233]

Returns the name of the table that stores this foreign key.

- **Access** public

The function ForeignKey::getName() [line 78]

Returns the name of the foreign key

- **Access** public

The function ForeignKey::getOnDelete() [line 212]

Returns the configured SQL 'OnDelete' action for the foreign key.

- **Access** public

The function ForeignKey::getOnUpdate() [line 188]

Returns the configured SQL 'OnUpdate' action for the foreign key.

- **Access** public

The function ForeignKey::getReferencedColumn() [line 141]

Returns the name of the referenced (parent) column.

- **Access** public

The function ForeignKey::getReferencedColumnInstance() [line 300]

Returns an instance of the referenced column

- **Access** public

The function ForeignKey::getReferencedController() [line 165]

Returns the name of the referenced MVC controller responsible for the management of the referenced domain model.

- **Access** public

The function ForeignKey::getReferencedTable() [line 120]

Returns the physical name of the referenced table.

- **Access** public

The function ForeignKey::getReferencedTableInstance() [line 289]

Returns an instance of the referenced Table.

- **Access** public

void function ForeignKey::getSelect() [line 277]

Gets the name of the foreign table's column which should be displayed in a drop-

down in place of the actual foreign key value.

- **Access** public

The function ForeignKey::getSelectedColumnInstance() [line 337]

Returns the foreign (parent) column instance which contains a select="true" configuration in persistence.xml. If a column is not explicitly set, the referenced column instance is returned.

- **Access** public

The function ForeignKey::getType() [line 99]

Returns the data type being stored in the foreign key column.

- **Access** public

void function ForeignKey::setFkColumn(\$columnName) [line 244]

Function Parameters:

- *\$columnName* **\$columnName** The physical column name

The name of the column which stores this foreign key

- **Access** public

void function ForeignKey::setFkTable(\$tableName) [line 223]

Function Parameters:

- *\$tableName* **\$tableName** The physical name of the foreign key table.

Sets the physical name of the foreign key table.

- **Access** public

void function ForeignKey::setName(\$name) [line 68]

Function Parameters:

- *\$name* **\$name** The foreign key name

Sets the name of the foreign key. This is the actual name given to the foreign key according to the database server.

- **Access** public

void function ForeignKey::setOnDelete(\$action) [line 201]

Function Parameters:

- *\$action* **\$action** The action to perform when the foreign (parent) key is deleted.
(NO_ACTION|RESTRICT|CASCADE|SET_NULL|SET_DEFAULT)

Sets the SQL 'onDelete' action which is invoked when the foreign (parent) key is updated.

- **Access** public

void function ForeignKey::setOnUpdate(\$action) [line 178]

Function Parameters:

- *\$action* **\$action** The action to perform
(NO_ACTION|RESTRICT|CASCADE|SET_NULL|SET_DEFAULT)

Sets the SQL 'OnUpdate' action for the foreign key. This action is performed automatically by the database server anytime the primary key in the foreign (parent) table is updated.

- **Access** public

void function ForeignKey::setReferencedColumn(\$columnName) [line 131]

Function Parameters:

- *\$columnName* **\$columnName** The referenced column name

Sets the name of the referenced (parent) column.

- **Access** public

void function ForeignKey::setReferencedController(\$controller) [line 154]

Function Parameters:

- *\$controller* **\$controller** The controller responsible for the management of the referenced domain model

Sets the name of the referenced MVC controller responsible for the management of the referenced domain model.

- **Access** public

void function ForeignKey::setReferencedTable(\$tableName) [line 110]

Function Parameters:

- *\$tableName* **\$tableName** The name of the referenced (parent) table.

Sets the name of the referenced table.

- **Access** public

void function ForeignKey::setSelect(\$columnName) [line 266]

Function Parameters:

- *\$columnName* **\$columnName** The foreign column name

Sets the name of the foreign table's column which should be displayed in a drop-down in place of the actual foreign key value.

- **Access** public

void function ForeignKey::setType(\$type) [line 89]

Function Parameters:

- *\$type* **\$type** The data type of the value being stored in the foreign key column

Sets the foreign key value data type.

- **Access** public

Class PersistenceRenderer

[line 31]

AgilePHP :: PersistenceRenderer Handles rendering persisted data to HTML

- **Package** com-makeabyte-agilephp-persistence
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

True function PersistenceRenderer::isBit(\$table, \$columnName) *[line 107]*

Function Parameters:

- *\$table* **\$table** The 'Table' object which contains the column
- *\$columnName* **\$columnName** The name of the column to search

Returns boolean response based on the configured 'type' attribute for the specified column.

- **Static**
- **Access** public

An function PersistenceRenderer::toFormElement(\$table, \$columnName, \$value, [\$var = null]) [*line 45*]

Function Parameters:

- *\$tableElement* **\$table** SimpleXML table element containing the column
- *\$columnName* **\$columnName** The name of the column to render as an HTML input form element
- *\$value* **\$value** An optional value to assign to the input element
- *\$var* **\$var** A 'special' variant variable used in rendering algorithm. The variable has a different meaning for each of the contexts it is invoked.

Creates an HTML input form element for the specified column.

- **Static**
- **Access** public

Class Table

[*line 31*]

AgilePHP :: Table Represents a data table in the AgilePHP persistence component.

- **Package** com-makeabyte-agilephp-persistence
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void* function Table::__construct(\$table) [*line 43*]

Function Parameters:

- *SimpleXMLElement* **\$table**

- **Access** public

void function Table::addColumn(\$column) [*line 225*]

Function Parameters:

- [Column](#) **\$column** The Column instance to push onto the stack

Pushes a new column to the table's column stack.

- **Access** public

unknown_type function Table::getColumnNameByProperty(\$property) [*line 320*]

Function Parameters:

- *\$property* **\$property**

Looks up a 'Column' name by its persistence.xml 'property' attribute.

- **Access** public

Array function Table::getColumns() [*line 246*]

Returns an array of Column instances which are configured for the table.

- **Access** public

The function Table::getDescription() [line 160]

Returns the table description

- **Access** public

The function Table::getDisplay() [line 117]

Returns the friendly display name for the table.

- **Access** public

The function Table::getDisplayNameByProperty(\$property) [line 344]

Function Parameters:

- *\$property* **\$property** The name of the 'property' to retrieve the display name for

Returns the 'display' attribute value as configured in persistence.xml for the specified 'property' value.

- **Access** public

An function Table::getForeignKeyColumns() [line 276]

Returns a list of foreign key columns in the current table.

- **Access** public

An function Table::getForeignKeyColumnsByKey(\$name) [*line 302*]

Function Parameters:

- **\$name** **\$name** The name of the foreign key

Returns a list of all foreign keys which share the same 'name' attribute value

- **Access** public

The function Table::getModel() [*line 94*]

Returns the name of the domain model responsible for the table

- **Access** public

An function Table::getModelInstance() [*line 427*]

Returns an instance of the model as configured in persistence.xml for this 'Table'.

- **Access** public

The function Table::getName() [*line 73*]

Returns the name of the database table

- **Access** public

An function Table::getPrimaryKeyColumns() [line 261]

Returns a list of primary key columns in the current table.

- **Access** public

void function Table::getSelectableColumns() [line 286]

- **Access** public

True function Table::getValidate() [line 138]

Returns the boolean field used to toggle automatic data validation.

- **Access** public

The function Table::getViewDisplayName() [line 440]

Returns a name suitable for display to end users. If a friendly name has been configured for this table (by providing a display attribute value in persistence.xml for the Column), then this value is returned, otherwise the table name attribute value is returned instead.

- **Access** public

True function Table::hasBlobColumn() [line 378]

Returns boolean response based on the presence of a column which is configured in persistence.xml as a blob data type.

- **Access** public

True function Table::hasColumns() [*line 236*]

Returns boolean response based on the presence of one or more Column instances.

- **Access** public

True function Table::hasForeignKey() [*line 394*]

Returns boolean response based on the presence of a column which is configured in persistence.xml as a foreign key.

- **Access** public

True function Table::hasForeignKeyReferences() [*line 410*]

Returns boolean response based on whether or not there are any other tables which have foreign keys referencing this table.

- **Access** public

True function Table::isIdentity() [*line 181*]

Returns true if the table is configured as an AgilePHP Identity table, false otherwise.

- **Access** public

unknown_type function Table::isSession() [*line 203*]

Returns boolean indicator based on whether or not this is a table responsible for AgilePHP session table.

- **Access** public

True function Table::isVisible(\$property) [*line 362*]

Function Parameters:

- *\$property* **\$property** The domain object model's property name

Returns boolean response based on the 'visible' attribute in persistence.xml for the column which contains the specified \$property.

- **Access** public

void function Table::setColumns(\$columns) [*line 214*]

Function Parameters:

- *array* **\$columns** An array of Column instances which belong to this table

Sets the array of columns for the table

- **Access** public

void function Table::setDescription(\$description) [line 150]

Function Parameters:

- *\$description* **\$description** A short description of the table

Sets the description field for the table. This field is used by visual elements which reference the tables description.

- **Access** public

void function Table::setDisplay(\$display) [line 107]

Function Parameters:

- *\$display* **\$display** The friendly display name for the table

Text field used to configure a 'display name'. If a value is specified this is the name that is used in visual elements such as forms and result lists when referencing the table.

- **Access** public

void function Table::setIsIdentity(\$boolean) [line 171]

Function Parameters:

- *\$boolean* **\$boolean** True if the table is responsible for AgilePHP Identity persistence.

Sets the boolean field used to identify an AgilePHP Identity table.

- **Access** public

True function Table::setIsSession(\$boolean) [line 192]

Function Parameters:

- *\$boolean* **\$boolean** True if the table is responsible for AgilePHP Session persistence.

Sets the boolean field used to identity an AgilePHP Session table.

- **Access** public

void function Table::setModel(\$model) [line 84]

Function Parameters:

- *\$model* **\$model** The name of the domain model

Sets the name of the domain model responsible for the table

- **Access** public

void function Table::setName(\$name) [line 63]

Function Parameters:

- *\$name* **\$name** The name of the database table

Sets the table name

- **Access** public

unknown_type function Table::setValidate(\$boolean) [*line 128*]

Function Parameters:

- *\$boolean* **\$boolean** True to enable validation, false to disable. Default is True.

Boolean field used to toggle automatic data validation for the table

- **Access** public

Package com-makeabyte-agilephp-persistence-dialect Procedural Elements

MySQLDialect.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework
Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-persistence-dialect

SQLDialect.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-persistence-dialect

SQLiteDialect.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-persistence-dialect

Package com-makeabyte-agilephp-persistence-dialect Classes

Class MySQLDialect

[line 31]

AgilePHP :: MySQLDialect Handles MySQL specific queries

- **Package** com-makeabyte-agilephp-persistence-dialect
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc.

Constructor *void* function MySQLDialect::__construct(\$db) [line 39]

Function Parameters:

- [*Database*](#) **\$db** The database to use

Inititalize MySQLDialect

- **Access** public

void function MySQLDialect::create() [line 70]

Creates the database as defined in persistence.xml

- **TODO** Add engine and charset attributes to persistence.xml 'table' element and assign values from xml definitions. Also need support for dynamic setting of unique key, fulltext, key, index, etc...
- **Throws** AgilePHP_PersistenceException
- **Access** public

void function MySQLDialect::drop() [line 174]

Drop/destroy the database.

- **Access** public

Class SQLDialect

[line 31]

AgilePHP :: SQLDialect Defines contract for dealing with vendor specific SQL queries

- **Package** com-makeabyte-agilephp-persistence-dialect
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

void function SQLDialect::create() [line 33]

- **Access** public

void function SQLDialect::drop() [*line 34*]

- **Access** public

Class SQLiteDialect

[*line 31*]

AgilePHP :: SQLiteDialect Responsible for SQLite specific database operations

- **Package** com-makeabyte-agilephp-persistence-dialect
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void* function SQLiteDialect::__construct(\$db) [*line 33*]

Function Parameters:

- [*Database*](#) \$db

- **Access** public

unknown_type function SQLiteDialect::create() [*line 44*]

```
CREATE TABLE "phpunit2" ("id" INTEGER NOT NULL ,  
    "fld1" VARCHAR NOT NULL , "fld2" VARCHAR DEFAULT  
    'This is nullable', "fld3" VARCHAR NOT NULL , "fld4" TEXT  
    NOT NULL , PRIMARY KEY ("id", "fld1"))
```


- **Access** public

void function SQLiteDialect::drop() [*line 122*]

Delete the database.

- **Access** public

void function SQLiteDialect::truncate(\$model) [*line 111*]

Function Parameters:

- **\$model**

- **Access** public

Package com-makeabyte-agilephp-scope

Procedural Elements

ApplicationScope.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework
Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-scope

RequestScope.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-scope

Session.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-scope

SessionScope.php

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

- **Package** com-makeabyte-agilephp-scope

Package com-makeabyte-agilephp-scope

Classes

Class ApplicationScope

[line 31]

AgilePHP :: ApplicationScope Uses the server temp directory to store PHP application data

- **Package** com-makeabyte-agilephp-scope
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc.

void function ApplicationScope::clear() [line 109]

Clears the ApplicationScope state

- **Access** public

void function ApplicationScope::destroy() [line 120]

Clears the ApplicationScope store and deletes the temp file associated with the application.

- **Access** public

The function `ApplicationScope::get($key)` [line 84]

Function Parameters:

- `$key` **\$key** The key to retrieve from the ApplicationScope store.

Returns the value corresponding to the specified key.

- **Access** public

An function `ApplicationScope::getInstance([$appName = null])` [line 68]

Function Parameters:

- `$appName` **\$appName** An optional application name. Defaults to the HTTP HOST header.

Returns a singleton instance of ApplicationScope

- **Static**
- **Access** public

`void` function `ApplicationScope::persist()` [line 141]

Persists the ApplicationScope state

- **Access** public

void function ApplicationScope::set(\$key, \$value) [line 99]

Function Parameters:

- **\$key \$key** The variable name
- **\$value \$value** The variable value

Sets an ApplicationScope variable

- **Access** public

void function ApplicationScope::__destruct() [line 131]

Persist the ApplicationScope state to disk

- **Access** public

Class RequestScope

[line 33]

AgilePHP :: RequestScope Stores PHP \$_POST variables, performs sanitizing, and has built in support to guard against CSFR attacks using a double cookie submit approach.

- **Package** com-makeabyte-agilephp-scope
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

The function RequestScope::createToken() [line 140]

Creates a 20-30 character token used to guard against CSFR attacks.

- See http://en.wikipedia.org/wiki/Cross-site_request_forgery
- Access public

The function RequestScope::get(\$key) [line 98]

Function Parameters:

- \$key **\$key** The key/index of the variable

Returns a store variable.

- Access public

Singleton function RequestScope::getInstance() [line 72]

Returns singleton instance of RequestScope

- Static
- Access public

Returns function RequestScope::getParameters() [line 129]

Returns the variable store.

- Access public

The function RequestScope::getSanitized(\$key) [line 116]

Function Parameters:

- **\$key \$key** The key/index of the variable

Returns a sanitized store variable which protects against SQL injection, XSS and XSFR attacks. The variable value is passed first through strip_tags, followed by addslashes and finally htmlspecialchars. Use htmlspecialchars_decode to decode the sanitized value.

- See http://en.wikipedia.org/wiki/Cross-site_scripting
- Access public

void function RequestScope::invalidate() [line 171]

Expires request token cookie.

- Access public

The function RequestScope::sanitize(\$data) [line 183]

Function Parameters:

- **\$data \$data** The data to sanitize

Sanitizes the specified data by running it through htmlspecialchars, addslashes, and strip_tags.

- Access public

void function RequestScope::set(\$key, \$value) [line 87]

Function Parameters:

- **\$key \$key** The key/index of for the variable
- **\$value \$value** The variable value

Sets a store variable.

- **Access** public

Class Session

[line 32]

AgilePHP :: Session Session domain model object.

- **Package** com-makeabyte-agilephp-scope
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

Constructor *void function Session::__construct() [line 38]*

- **Access** public

Timestamp function Session::getCreated() [line 98]

Returns the timestamp when the session was created

- **Access** public

void function Session::getData() [line 77]

Returns the serialized session data

- **Access** public

Session function Session::getId() [line 56]

Returns the session id

- **Access** public

void function Session::setCreated(\$timestamp) [line 88]

Function Parameters:

- *\$dateTime* **\$timestamp** Timestamp indicating when the session was created

Timestamp indicating when the session was created

- **Access** public

void function Session::setData(\$data) [line 67]

Function Parameters:

- **\$data \$data** Serialized session data

Stores the serialized session

- **Access** public

void function Session::setId(\$id) [line 46]

Function Parameters:

- **\$id \$id** The session id

Sets the session id

- **Access** public

Class SessionScope

[line 30]

AgilePHP :: SessionScope

- **Package** com-makeabyte-agilephp-scope
- **Author** Jeremy Hahn
- **Version** 0.1a
- **Copyright** Make A Byte, inc

void function SessionScope::clear() [line 163]

Clears the current Session.

- **Access** public

void function SessionScope::destroy() [line 179]

Clears the SessionScope store and deletes the Session ActiveRecord from the database.

- **Access** public

The function SessionScope::get(\$key) [line 135]

Function Parameters:

- **\$key \$key** The variable's key/name

Returns the value corresponding to the specified key stored in the current Session.

- **Access** public

Singleton function SessionScope::getInstance() [line 79]

Returns a singleton instance of SessionScope

- **Static**
- **Access** public

void function SessionScope::getSession() [line 93]

Returns the session domain model object which maintains the id and data for the current Session ActiveRecord.

- **Access public**

The function SessionScope::getSessionId() [line 103]

Returns the session id for the current Session.

- **Access public**

True function SessionScope::isPersisted() [line 193]

Returns boolean flag indicating whether or not the current session data is persisted.

- **Access public**

void function SessionScope::persist() [line 224]

Persists a serialized instance of the current Session.

- **Access public**

void function SessionScope::set(\$key, \$value) [line 151]

Function Parameters:

- **\$key \$key** The variable name
- **\$value \$value** The variable value

Sets a new Session variable.

- **Access** public

void function SessionScope::setSessionId(\$id) [line 113]

Function Parameters:

- **\$id**

Sets the session id and attempts to restore a previously persisted Session.

- **Access** public

void function SessionScope::__destruct() [line 207]

Persist the Session data state to database just before the object is destroyed.

- **Access** public

Appendices

Appendix A - Class Trees

Package com-makeabyte-agilephp-annotation

AnnotatedClass

- ReflectionClass
 - [AnnotatedClass](#)

AnnotatedMethod

- ReflectionMethod
 - [AnnotatedMethod](#)

AnnotatedProperty

- ReflectionProperty
 - [AnnotatedProperty](#)

AnnotationParser

- [AnnotationParser](#)

Package com-makeabyte-agilephp-annotation-annotations

AroundInvoke

- [AroundInvoke](#)

Interceptor

- [Interceptor](#)

RemoteMethod

- [RemoteMethod](#)

Package com-makeabyte-agilephp

AgilePHP

- [AgilePHP](#)

AgilePHP_Exception

- [AgilePHP_Exception](#)
 - [AgilePHP_AccessDeniedException](#)
 - [AgilePHP_AnnotationException](#)
 - [AgilePHP_InterceptionException](#)
 - [AgilePHP_NotLoggedInException](#)
 - [AgilePHP_PersistenceException](#)
 - [AgilePHP_RemotingException](#)

Annotation

- [Annotation](#)

Crypto

- [Crypto](#)

Form

- [Form](#)

i18n

- [i18n](#)

Identity

- [Identity](#)

Interception

- [Interception](#)

Logger

- [Logger](#)

MVC

- [MVC](#)

PersistenceManager

- [PersistenceManager](#)

Remoting

- [Remoting](#)

Scope

- [Scope](#)

Upload

- [Upload](#)

Validator

- [Validator](#)

Package com-makeabyte-agilephp-persistence

BasePersistence

- [BasePersistence](#)
 - [MySQLDialect](#)
 - [SQLiteDialect](#)

Column

- [Column](#)

Database

- [Database](#)

ForeignKey

- [ForeignKey](#)

PersistenceRenderer

- [PersistenceRenderer](#)

Table

- [Table](#)

Package com-makeabyte-agilephp-persistence-dialect

MySQLDialect

- [BasePersistence](#) (Different package)
 - [MySQLDialect](#)

SQLDialect

- [SQLDialect](#)

SQLiteDialect

- [BasePersistence](#) (Different package)
 - [SQLiteDialect](#)

Package com-makeabyte-agilephp-interception-interceptors

In

- [In](#)

Restrict

- [Restrict](#)

Package com-makeabyte-agilephp-interception

InterceptorProxy

- [InterceptorProxy](#)

InvocationContext

- [InvocationContext](#)

Package com-makeabyte-agilephp-scope

ApplicationScope

- [ApplicationScope](#)

RequestScope

- [RequestScope](#)

Session

- [Session](#)

SessionScope

- [SessionScope](#)

Package com-makeabyte-agilephp-mvc

BaseController

- [BaseController](#)
 - [BaseModelController](#)
 - [BaseModelXmlController](#)
 - [BaseModelXslController](#)
 - [BaseModelActionController](#)

BaseRenderer

- [BaseRenderer](#)
 - [AJAXRenderer](#)
 - [PHTMLRenderer](#)
 - [XSLTRenderer](#)

Package com-makeabyte-agilephp-identity

IdentityManager

- [IdentityManager](#)

IdentityModel

- [IdentityModel](#)

Role

- [Role](#)

User

- [User](#)

Appendix B - README/CHANGELOG/INSTALL

README

AgilePHP Framework :: The Rapid "for developers" PHP5 framework
Copyright (C) 2009 Make A Byte, inc

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <<http://www.gnu.org/licenses/>>.

----- ** Extensions **

--with-apxs2 (Dynamic apache support)
--with-pdo-<database type> (required for each database you want to use)
--enable-mbstring (XML/XSLT controllers use this to convert data to entities)
--with-gettext (i18n Internationalization)
--with-xsl (Controllers involved with XSLT)
--with-mcrypt (Crypto encrypt/decrypt methods)

----- ** PHPUnit Testing **

To use the testing packages, and TDD features, install PHP Unit 3 via PEAR.

pear channel-discover pear.phpunit.de
pear install phpunit/PHPUnit

----- ** TODO **

> Add constructor argument support to InterceptorProxy
> Iron out RequestScope CSFR issues in regards to destroying session
> Persistence framework needs to support many-to-many relationships
> Need to revisit PersistenceRenderer and make an easy way to
allow different client side rendering libraries like jquery, dojo, extjs, etc (adapter?)
> Add calendar, spinner, date picker, and other widgets to persistence rendering
> AJAX Push / Comet (PHP socket daemon?)
> Create dialect classes for the following database servers:
pgsql
firebird
informix

oracle
dblib
ibm

> Finish Form::toJSON method (this should probably use AJAXRenderer)

Appendix D - Todo List

In Package com-makeabyte-agilephp-persistence-dialect

In [MySQLDialect::create\(\)](#)

- Add engine and charset attributes to persistence.xml 'table' element and assign values from xml definitions. Also need support for dynamic setting of unique key, fulltext, key, index, etc...

Index

A

AnnotatedProperty::getAnnotations()	104
<i>Returns all property annotations. If a name is specified only annotations which match the specified name will be returned, otherwise all annotations are returned.</i>	
AnnotatedProperty::getDeclaringClass()	105
<i>Gets the parent class as an AnnotatedClass</i>	
AnnotatedProperty::getAnnotation()	104
<i>Gets an annotation instance by name. If the named annotation is found more than once, an array of annotations are returned.</i>	
AnnotatedProperty	103
<i>AgilePHP :: AnnotatedProperty</i>	
<i>Extends the PHP ReflectionClass to provide details about property level annotations.</i>	
AnnotatedMethod::isAnnotated()	103
<i>Returns boolean indicator based on the presence of any annotations.</i>	
AnnotatedProperty::hasAnnotation()	105
<i>Checks the property for the presence of the specified annotation.</i>	
AnnotatedProperty::isAnnotated()	105
<i>Returns boolean indicator based on the presence of any annotations.</i>	
AnnotationParser::getInstance()	107
AnnotationParser::getClassAnnotationsAsArray()	106
<i>Returns an array of class level annotations</i>	
AnnotationParser::getClassAnnotations()	106
<i>Returns an array of class level annotations or false if no class level annotations are present.</i>	
AnnotationParser	106
<i>AgilePHP :: AnnotationParser</i>	
<i>Responsible for parsing and returning annotation details about PHP classes.</i>	
AnnotatedMethod::hasAnnotation()	103
<i>Checks the method for the presence of the specified annotation.</i>	
AnnotatedMethod::getDeclaringClass()	102
<i>Gets the parent class as an AnnotatedClass</i>	
AnnotatedClass::getProperties()	100
<i>Returns an array of AnnotatedProperty objects; one for each property in the class which contains an annotation.</i>	
AnnotatedClass::getMethods()	100
<i>Returns an array of AnnotatedMethod objects, one for each method in the class which contains an annotation.</i>	
AnnotatedClass::getMethod()	99
<i>Returns an AnnotatedMethod instance for the specified method name.</i>	
AnnotatedClass::getAnnotations()	99
<i>Returns all class level annotations. If a name is specified only annotations which match the specified name will be returned, otherwise all annotations are returned.</i>	

AnnotatedClass::getProperty()	100
<i>Returns an AnnotatedProperty instance for the specified property name.</i>	
AnnotatedClass::hasAnnotation()	100
<i>Checks the class for the presence of the specified annotation.</i>	
AnnotatedMethod::getAnnotations()	102
<i>Returns all method annotations. If a name is specified only annotations which match the specified name will be returned, otherwise all annotations are returned.</i>	
AnnotatedMethod::getAnnotation()	102
<i>Gets an annotation instance by name. If the named annotation is found more than once, an array of annotations are returned.</i>	
AnnotatedMethod	101
<i>AgilePHP :: AnnotatedMethod</i>	
<i>Extends the PHP ReflectionClass to provide details about method level annotations.</i>	
AnnotatedClass::isAnnotated()	101
<i>Returns boolean indicator based on whether or not there are any annotations present.</i>	
AnnotationParser::getMethodAnnotations()	107
<i>Returns an array of method level annotations or false if no annotations are found for the specified method.</i>	
AnnotationParser::getMethodAnnotationsAsArray()	107
<i>Returns an array of method level annotations for the specified class</i>	
ApplicationScope::clear()	262
<i>Clears the ApplicationScope state</i>	
ApplicationScope	262
<i>AgilePHP :: ApplicationScope</i>	
<i>Uses the server temp directory to store PHP application data</i>	
ApplicationScope.php	258
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
AJAXRenderer::setOutput()	168
<i>Sets the desired output type.</i>	
ApplicationScope::destroy()	262
<i>Clears the ApplicationScope store and deletes the temp file associated with the application.</i>	
ApplicationScope::get()	263
<i>Returns the value corresponding to the specified key.</i>	
ApplicationScope::destruct()	264
<i>Persist the ApplicationScope state to disk</i>	
ApplicationScope::set()	264
<i>Sets an ApplicationScope variable</i>	
ApplicationScope::persist()	263
<i>Persists the ApplicationScope state</i>	
ApplicationScope::getInstance()	263
<i>Returns a singleton instance of ApplicationScope</i>	
AJAXRenderer::renderNoHeader()	167
<i>Renders the specified data according to \$output without sending an HTTP content-type header.</i>	
AJAXRenderer::renderNoFormatNoHeader()	167
<i>Renders the specified data without sending it through any of the AJAXRenderer internal formatting/converting methods. In addition, no HTTP content-type header is added to the response.</i>	
AnnotationParser::parse()	108

<i>Breaks the class file into PHP tokens and extracts all interface, class, method, and property level annotations.</i>	
AnnotationParser::getSourceCode()	108
<i>Returns the PHP file content to be parsed.</i>	
AnnotationParser::getPropertyAnnotationsAsArray()	108
<i>Returns an array of property level annotations for the specified class</i>	
AnnotationParser::getPropertyAnnotations()	107
<i>Returns an array of property level annotations or false if no annotations are present for the specified property.</i>	
AroundInvoke.php	111
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
AroundInvoke	114
<i>AgilePHP :: AroundInvoke</i>	
<i>Annotation which causes #@Interceptor methods to be invoked</i>	
AJAXRenderer::renderNoFormat()	167
<i>Renders the specified data without sending it through any of the AJAXRender internal formatting/convertng methods. An appropriate HTTP content-type header is added to the response.</i>	
AJAXRenderer::render()	166
<i>Renders the specified PHP data according to \$output. The appropriate content-type to the HTTP response header.</i>	
AJAXRenderer	166
<i>AgilePHP :: MVC AJAXRenderer</i>	
<i>Renders data to JSON and XML with appropriate content-type header.</i>	
AJAXRenderer.php	157
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
AnnotatedClass::getAnnotation()	99
<i>Gets an annotation instance by name. If the named annotation is found more than once, an array of annotations are returned.</i>	
AnnotatedClass	98
<i>AgilePHP :: AnnotatedClass</i>	
<i>Extends the PHP ReflectionClass to provide details about class level annotations.</i>	
AgilePHP::isInDebugMode()	20
<i>Whether or not AgilePHP framework is running in debug mode.</i>	
AgilePHP::initMVC()	20
<i>Initializes a default Model-View-Control component without reading agilephp.xml configuration.</i>	
AgilePHP::getXmlConfiguration()	20
<i>Returns the agilephp.xml configuration as a SimpleXMLElement.</i>	
AgilePHP::getWebRoot()	20
<i>Returns the fully qualified path to the base directory of the web application.</i>	
AgilePHP::setAppName()	21
<i>Sets the name of the AgilePHP web application.</i>	
AgilePHP::setDebugMode()	21
<i>Enables or disables AgilePHP framework debug mode.</i>	
AgilePHP::setFrameworkRoot()	22
<i>Sets the full system path to the location of the AgilePHP framework. The given path is appended to the current php.ini include_path configuration.</i>	
AgilePHP::setDocumentRoot()	22

<i>Sets the relative path to the web application from the server's document root.</i>	
AgilePHP::setDisplayPhpErrors()	22
<i>By default PHP hides errors on production servers. Setting this to true enables PHP 'display_errors', sets 'error_reporting' to 'E_ALL'.</i>	
AgilePHP::setDefaultTimezone()	21
<i>Calls PHP date_default_timezone_set function to set the current timezone.</i>	
AgilePHP::getRequestBase()	19
<i>Returns the base action url used to communicate with the AgilePHP MVC component. Defaults to the name of the script which initializes the framework.</i>	
AgilePHP::getMVC()	19
<i>Accessor method for Model-View-Control component. The MVC component this method returns is created as follows:</i>	
AgilePHP::disableInterceptors()	17
<i>Disables interceptors.</i>	
AgilePHP::addInterception()	17
<i>Adds an Interception to the interceptions stack</i>	
AgilePHP	17
<i>AgilePHP :: Core Framework Class</i>	
Annotation.php	4
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework Copyright (C) 2009 Make A Byte, inc</i>	
AgilePHP::enableInterceptors()	18
<i>Enables the use of interceptors.</i>	
AgilePHP::getAppName()	18
<i>Returns the name of the AgilePHP web application.</i>	
AgilePHP::getInterceptions()	19
<i>Returns an array of Interceptions which have been loaded into the framework</i>	
AgilePHP::getFrameworkRoot()	19
<i>Gets the full system path to the location where AgilePHP resides.</i>	
AgilePHP::getFramework()	18
<i>Factory method which is used to instantiate a singleton instance of the AgilePHP framework.</i>	
AgilePHP::getDocumentRoot()	18
<i>Returns the relative path to the web application from the server's document root.</i>	
AgilePHP::setRequestBase()	23
<i>Sets the base action url used to communicate with the AgilePHP MVC component.</i>	
AgilePHP::setWebRoot()	23
<i>Sets the fully qualified path to the base directory of the web application.</i>	
Annotation::getPropertiesAsArray()	29
<i>Returns an array of property level annotations for the specified class/property.</i>	
Annotation::getMethodsAsArray()	29
<i>Returns an array of method level annotations for the specified class/method.</i>	
Annotation::getMethod()	29
<i>Static factory method used to retrieve AnnotatedMethod instances.</i>	
Annotation::getClassAsArray()	28
<i>Returns an array of class level annotations for the specified class.</i>	
Annotation::getProperty()	30
<i>Static factory method used to retrieve AnnotatedProperty instances.</i>	
Annotation::hasAnnotations()	30
<i>Returns true if the specified class contains and class, method, or</i>	

property level annotations.	
AnnotationParser.php	97
AgilePHP Framework :: The Rapid "for developers" PHP5 framework Copyright (C) 2009 Make A Byte, inc	
AnnotatedProperty.php	96
AgilePHP Framework :: The Rapid "for developers" PHP5 framework Copyright (C) 2009 Make A Byte, inc	
AnnotatedMethod.php	95
AgilePHP Framework :: The Rapid "for developers" PHP5 framework Copyright (C) 2009 Make A Byte, inc	
AnnotatedClass.php	94
AgilePHP Framework :: The Rapid "for developers" PHP5 framework Copyright (C) 2009 Make A Byte, inc	
Annotation::getClass()	28
Static factory method used to retrieve AnnotatedClass instances.	
Annotation	27
AgilePHP :: Annotation Enables the use of annotations in PHP.	
AgilePHP AccessDeniedException	24
Thrown when a user attempts to access content which they do not have permission to view.	
AgilePHP::destruct()	24
AgilePHP::stopClock()	24
Stops the timer and returns the elapsed time between startClock() and stopClock().	
AgilePHP::startClock()	23
Starts a timer. Useful for measuring how long a particular operation takes.	
AgilePHP AnnotationException	24
Exceptions thrown by the Annotation package	
AgilePHP Exception	25
Base AgilePHP exception class	
AgilePHP RemotingException	27
Handles all remoting exceptions. Output is returned in JSON format with an application/json HTTP header.	
AgilePHP PersistenceException	26
Exceptions thrown by the Persistence/ORM framework	
AgilePHP NotLoggedInException	26
Thrown when a user is not logged in and requests secure content	
AgilePHP InterceptionException	26
Exceptions thrown by the Interception package	
AgilePHP.php	2
AgilePHP Framework :: The Rapid "for developers" PHP5 framework Copyright (C) 2009 Make A Byte, inc	

B

BasePersistence::beginTransaction()	203
Begins a transaction	
BasePersistence::close()	203
Closes the connection to the database.	
BasePersistence::\$transactionInProgress	203

BasePersistence::\$pdo	203
BasePersistence::\$database	202
BasePersistence::\$model	202
BasePersistence::commit()	204
<i>Commits an already started transaction.</i>	
BasePersistence::compare()	204
<i>Compares domain model object \$a with \$b.</i>	
BasePersistence::find()	205
<i>Attempts to locate the specified model by primary key value.</i>	
BasePersistence::getAvailableDrivers()	206
<i>Returns a list of drivers available to PDO.</i>	
BasePersistence::execute()	205
<i>Executes a prepared statement (with parameters)</i>	
BasePersistence::exec()	205
<i>Executes an SQL statement and returns the number of rows affected by the query.</i>	
BasePersistence::delete()	204
<i>Deletes a persisted domain model object (ActiveRecord)</i>	
BasePersistence	202
<i>AgilePHP :: BasePersistence</i>	
<i>Base persistence class responsible for all database/ORM actions</i>	
BasePersistence.php	195
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
BaseModelXslController::getModelAsReadOnlyXSL()	186
<i>Returns an XSL stylesheet used for read-only (this is the Read in CRUD).</i>	
BaseModelXslController::getModelFormXSL()	186
<i>Returns an XSL stylesheet used for add and update actions. During an 'add' operation</i>	
BaseModelXslController	185
<i>AgilePHP :: MVC BaseModelXslController</i>	
<i>Provides base implementation for model xsl controllers.</i>	
BaseModelXmlController::getResultListAsXML()	185
<i>Returns a result set from the database as XML. The XML document is returned with the root node 'ResultList' containing an element named after the model which then contains each of the models properties and values as children.</i>	
BaseModelXmlController::getResultListAsPagedXML()	184
<i>Returns a paged result set from the database as XML (including foreign model instances with their</i>	
<i>primary keys set). The XML document is returned in the following format:</i>	
BaseModelXslController::getModelListXSL()	187
<i>Generates an XSL stylesheet from the domain model object's properties. Designed to be used in conjunction</i>	
<i>with getResultListAsPagedXML() to perform an XSLT transformation.</i>	
BaseModelXslController::getPaginationXSL()	187
<i>Returns an XSL stylesheet used for pagination.</i>	
BaseRenderer::render()	188
BaseRenderer::set()	189
<i>Helper method to allow controllers to set variables</i>	
<i>which are dumped to a view during rendering.</i>	
BaseRenderer::getStore()	188
<i>Returns the store which contains variable names with their associated</i>	
<i>values set by one or more controllers.</i>	
BaseRenderer::get()	188
<i>Returns the value for the specified key.</i>	

BaseRenderer	187
<i>AgilePHP :: MVC BaseRenderer</i>	
<i>Provides base rendering implementation</i>	
BasePersistence::getColumnnameForProperty()	206
<i>Returns the value of the 'name' attribute configured in persistence.xml for the specified \$propertyName.</i>	
BasePersistence::getDatabase()	207
<i>Returns the 'Database' object being used by the persistence framework.</i>	
BasePersistence::quote()	212
<i>Quotes a string so its theoretically safe to pass into a statement</i>	
BasePersistence::rollback()	212
<i>Rolls back a transaction.</i>	
BasePersistence::query()	212
<i>Executes a raw SQL query</i>	
BasePersistence::prepare()	211
<i>Prepares an SQL prepared statement</i>	
BasePersistence::persist()	211
<i>Persists a domain model object</i>	
BasePersistence::setDistinct()	213
<i>Adds an SQL distinct clause to 'find' operation.</i>	
BasePersistence::setMaxResults()	213
<i>Sets the 'maxResults' property value which is used during a 'find' operation which contains an empty model.</i>	
BasePersistence::truncate()	215
<i>Truncates the table for the specified domain model object</i>	
BasePersistence::destruct()	215
<i>Closes the connection to the database.</i>	
BasePersistence::toMutator()	214
<i>Creates a mutator method from the \$property parameter. The \$property will be returned with the prefix 'set' and the first letter of the property uppercased.</i>	
BasePersistence::toBigint()	214
<i>Converts the specified parameter to a bigint.</i>	
BasePersistence::toAccessor()	214
<i>Creates an accessor method from the \$property parameter. The \$property will be returned with the prefix 'get' and the first letter of the property uppercased.</i>	
BasePersistence::merge()	211
<i>Merges/updates a persisted domain model object</i>	
BasePersistence::isEmpty()	210
<i>Checks the model in use by the persistence framework for the presence of property values. If the model does not contain any values, it is considered empty.</i>	
BasePersistence::getIdentityTable()	208
<i>Returns a 'Table' object representing the table configured in persistence.xml as the AgilePHP 'Identity' table.</i>	
BasePersistence::getMaxResults()	208
<i>Returns the 'maxResults' property value which is used during a 'find' operation which contains an empty model.</i>	
BasePersistence::getIdentityModel()	207
<i>Returns an instance of the domain model object responsible for AgilePHP 'Identity' persistence.</i>	
BasePersistence::getDistinct()	207

Returns the 'distinct' column to use in an SQL SELECT statement if one has been defined.	
BasePersistence::getDisplayNameForColumn()	207
Returns a custom display name as configured in persistence.xml 'display' attribute for the specified column name. If the 'Column' name can not be matched, it is then compared against the 'Column' 'property' attribute value. If neither can be matched, the \$columnName is returned.	
BasePersistence::getPDOInstance()	208
Returns the PDO instance in use by the persistence framework.	
BasePersistence::getPropertyForColumn()	208
Returns the value of the 'property' attribute configured in persistence.xml for the specified \$columnName.	
BasePersistence::getTableByModelName()	210
Returns the 'Table' object responsible for the specified \$modelName.	
BasePersistence::getTableByName()	210
Returns a 'Table' object by its name as configured in persistence.xml	
BasePersistence::getTableByModel()	209
Returns the 'Table' object which is mapped to the specified 'Model'.	
BasePersistence::getSessionTable()	209
Returns the 'Table' object that represents the table configured in persistence.xml as an AgilePHP 'SessionScope' session table.	
BasePersistence::getSessionModel()	209
Returns an instance of the domain model object responsible for AgilePHP 'SessionScope' session persistence.	
BaseModelXmlController::getModelPersistenceAction()	184
Returns the type of action which the controllers should take when deciphering whether the operation is a persist or merge operation. If the primary key(s) contain a value, the action is assumed a merge. If the primary key(s) do not contain a value, the action is assumed persist.	
BaseModelXmlController::getModelAsFormXML()	183
Creates an XML document representing a model. If the 'id' parameter is set, a lookup is performed for the model with the specified 'id' and the XML is returned with data populated from the database result set.. If there is no 'id' set, the model's property nodes will be null. A custom controller and action can be set to modify default behavior.	
BaseModelActionController::modelSort()	173
Sorts the modelList according to the specified column name and direction.	
BaseModelActionController::persist()	173
Persists a new ActiveRecord defined by the model in the extension class.	
BaseModelActionController::modelList()	172
Performs a search for the model defined in the extension class and displays a paginated result list with edit and delete actions, as well as sortable column headers.	
BaseModelActionController::modelAction()	172
Handles the dispatching of action events from the view.	
BaseModelActionController::merge()	171
Merges the current ActiveRecord defined by the model in the extension class.	
BaseModelActionController::revokeAccess()	173
Throws AgilePHP_NotLoggedInException	
BaseModelController	174
AgilePHP :: MVC BaseModelController Provides base implementation for model controllers.	

BaseModelController::executeQuery()	175
<i>Executes the current SQL statement as defined in 'sql'.</i>	
BaseModelController::find()	175
<i>Search for an ActiveRecord for the model defined in the extension class.</i>	
BaseModelController::executeCountQuery()	175
<i>Executes an SQL count query for total number of records in the database.</i>	
BaseModelController::delete()	175
<i>Deletes the model ActiveRecord defined in the extension class. The model must have its primary key properties defined for this operation to succeed.</i>	
BaseModelController::createQuery()	174
<i>Sets the SQL statement to use when calling executeQuery.</i>	
BaseModelActionController::getXsltRenderer()	171
<i>Returns an instance of the MVC XSLTRenderer object.</i>	
BaseModelActionController::delete()	171
<i>Deletes the current ActiveRecord defined by the model in the extension class.</i>	
BaseModelXslController.php	162
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework Copyright (C) 2009 Make A Byte, inc</i>	
BaseRenderer.php	163
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework Copyright (C) 2009 Make A Byte, inc</i>	
BaseModelXmlController.php	161
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework Copyright (C) 2009 Make A Byte, inc</i>	
BaseModelController.php	160
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework Copyright (C) 2009 Make A Byte, inc</i>	
BaseModelActionController.php	159
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework Copyright (C) 2009 Make A Byte, inc</i>	
BaseController	168
<i>AgilePHP :: MVC BaseController Provides basic renderer implementations and defines 'index' abstract method.</i>	
BaseController::createCustomRenderer()	169
<i>Creates an instance of the specified custom renderer the controller will use to render views.</i>	
BaseController::index()	170
<i>Default controller action method.</i>	
BaseModelActionController	170
<i>AgilePHP :: MVC BaseModelActionController Provides a base implementation of a working model action controller.</i>	
BaseController::getRenderer()	170
<i>Returns the controllers view renderer.</i>	
BaseController::getBaseJS()	169
<i>Returns the raw JavaScript contents of the AgilePHP.js file and pre-configures the library with a default AgilePHP.debug, AgilePHP.MVC.controller, and AgilePHP.MVC.action value.</i>	
BaseController::createRenderer()	169
<i>Creates an instance of the specified renderer the controller will use to render views.</i>	
BaseModelController::getCount()	176
<i>Returns the total number of records in the database table</i>	
BaseModelController::getMaxResults()	176
<i>Returns the maximum number of results to retrieve. This translates</i>	

to an SQL LIMIT clause during SELECT operations.	
BaseModelController::setGroupBy()	180
Sets the SQL 'group by' clause.	
BaseModelController::setMaxResults()	180
Sets the maximum number of records to return in a result list	
BaseModelController::previousExists()	180
Returns boolean result based on whether or not a 'previous page' result is available during a pagination request.	
BaseModelController::persist()	180
Persists a new model ActiveRecord defined in the extension class.	
BaseModelController::nextExists()	179
Returns boolean result based on whether or not a 'next page' result is available during a pagination request.	
BaseModelController::setOrderBy()	181
Sets the SQL 'order by' clause.	
BaseModelController::setPage()	181
Sets the pagination page number and performs an SQL query to populate the 'resultList' and 'resultCount' properties with their appropriate values for the specified page.	
BaseModelController::toMutator()	183
Creates a mutator method from the \$property parameter. The \$property will be returned with the prefix 'set' and the first letter of the property uppercased.	
BaseModelXmlController	183
AgilePHP :: MVC BaseModelXmlController	
Provides base implementation for model xml controllers.	
BaseModelController::toAccessor()	182
Creates an accessor method from the \$property parameter. The \$property will be returned with the prefix 'get' and the first letter of the property uppercased.	
BaseModelController::setRestrictionsLogicOperator()	182
Sets the restriction operator (and/or) used in SQL WHERE clause.	
BaseModelController::setRestrictions()	181
Sets WHERE clause restrictions	
BaseModelController::merge()	179
Merges the model defined in the extension class. The model must have its primary key properties defined for this operation to succeed.	
BaseModelController::getTableName()	179
Returns the table name for the model defined in the extension class	
BaseModelController::getOrderBy()	177
Returns an associative array containing the current 'orderBy' clause. The results are returned with the name of the column as the index and the direction as the value.	
BaseModelController::getPage()	177
Returns the current page number	
BaseModelController::getNextResultList()	176
Sets the result list up with a 'next page' of a pagination request	
BaseModelController::getModelName()	176
Returns the model's class name.	
BaseModelController::getModel()	176
BaseModelController::getPageCount()	177
Returns the total number of pages. This is calculated by dividing the total number of records by 'maxResults'.	
BaseModelController::getPersistenceManager()	177
Returns an instance of PersistenceManager	

BaseModelController::getResultList()	178
<i>Returns a result set from the database. The result list is returned as an array of stdClass objects, each representing a row in the database.</i>	
BaseModelController::getResultListAsModels()	178
<i>Returns the result list as an array of populated model objects. If the model contains any foreign keys, by default the foreign model has only its primary keys which exist as foreign keys in the current model populated and the foreign model is added to its instance property in the current ActiveRecord.</i>	
BaseModelController::getResultCount()	178
<i>Returns the total number of records in the current result list.</i>	
BaseModelController::getQuery()	178
<i>Returns the current SQL query</i>	
BaseModelController::getPreviousResultList()	178
<i>Sets the result list up with a 'previous page' of a pagination request</i>	
BaseController.php	158
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework Copyright (C) 2009 Make A Byte, inc</i>	

C

Column::isBit()	219
<i>Returns boolean flag indicating whether or not the column is of the data type 'bit'. This is a *special* data type which is used to render HTML checkboxes by the presentation tier.</i>	
Column::isAutoIncrement()	218
<i>Returns boolean flag indicating whether or not this column is an AUTO_INCREMENT field.</i>	
Column::hasForeignKey()	218
<i>Returns boolean indicator based on whether or not the column is a foreign key field.</i>	
Column::getViewDisplayName()	218
<i>Returns the value which is returned to the presentation tier when displaying the name of the column. If persistence.xml contains a valid 'display' attribute, this value is returned, otherwise, the column name is returned.</i>	
Column::isForeignKey()	219
<i>Returns boolean indicator based on whether or not the column is a foreign key field.</i>	
Column::isPrimaryKey()	219
<i>Returns boolean flag indicating whether or not the column is a primary key</i>	
Column::isSortable()	220
<i>Returns boolean flag indicating whether or not the column is sortable.</i>	
Column::isSelectable()	220
<i>Returns boolean flag indicating whether or not the column can be used by child tables to render data in an HTML drop-down instead of their foreign key value(s)</i>	
Column::isRequired()	219
<i>Returns boolean flag indicating whether or not the column data is required</i>	
Column::getType()	218
<i>Returns the data type which describes the type of data which is being stored in the column/field.</i>	
Column::getProperty()	217
<i>Returns the name of the property in the domain model which the column maps</i>	
Column::getDescription()	216
<i>Returns the friendly description of the column.</i>	

Column::getDefault()	216
<i>Returns the default value assigned to the column data when a value has not been specified.</i>	
constructor Column::construct()	216
Column	215
<i>AgilePHP :: Column</i>	
<i>Represents a database table column in the AgilePHP persistence component.</i>	
Column::getDisplay()	216
<i>Returns the friendly display name for the column</i>	
Column::getForeignKey()	217
<i>Returns boolean indicator based on whether or not the column is a foreign key field.</i>	
Column::getName()	217
<i>Returns the name of the column in the physical database</i>	
Column::getModelPropertyName()	217
<i>Returns the name which is used to access/mutate model properties/fields. If a property attribute has been configured in persistence.xml for the column, the property value is returned, otherwise the column name is returned instead.</i>	
Column::getLength()	217
<i>Returns the length of the data being stored in the column</i>	
Column::isVisible()	220
<i>Returns boolean indicator based on the visibility of the column.</i>	
Column::setAutoIncrement()	220
<i>Marks the column as an AUTO_INCREMENT column.</i>	
constructor Database::construct()	226
Column::setVisible()	225
<i>Sets a boolean flag used to toggle the visibility of the column. This is used to hide a particular field (such as a primary key) from a form when its rendered.</i>	
Column::setType()	224
<i>Sets the data type which describes the type of data which is to be stored in the column/field.</i>	
Column::setSortable()	224
<i>Enables/disables sortable column headers,</i>	
constructor ForeignKey::construct()	231
constructor Table::construct()	239
constructor Session::construct()	267
constructor SQLiteDialect::construct()	255
constructor MySQLDialect::construct()	253
<i>Initialize MySQLDialect</i>	
Column::setSelectable()	224
<i>When set to true, this option allows tables who are a child table referencing this column, to render this columns data instead of its foreign key values.</i>	
Column::setRequired()	223
<i>Boolean flag indicating whether or not the column data is required</i>	
Column::setDisplay()	221
<i>Sets the friendly display name of the column. This is used by visual elements (such as forms) to show a user friendly name for the column.</i>	
Column::setDescription()	221
<i>Sets the friendly description of the column. This is used by visual elements to describe the column to the end user or website owner.</i>	
Column::setDefault()	221
<i>Sets the default value of the column data</i>	

Column::setForeignKey()	222
<i>Marks the column as a foreign key column.</i>	
Column::setLength()	222
<i>Sets the length of the data being stored in the column</i>	
Column::setProperty()	223
<i>Sets the name of the property in the domain model which the column maps</i>	
Column::setPrimaryKey()	223
<i>Marks this column as a primary key</i>	
Column::setName()	222
<i>Sets the name of the column in the physical database</i>	
constructor BasePersistence::construct()	203
Column.php	197
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
Crypto::getInstance()	35
<i>Returns a singleton instance of Crypto</i>	
Crypto::getDigest()	34
<i>Returns the hashed \$data. This operation requires either a valid configuration in agilephp.xml for the Crypto component or you must manually set the algorithm with a call to 'setAlgorithm()'.</i>	
Crypto::getAlgorithm()	34
<i>Returns the algorithm the Crypto component is configured to perform a hashing operation with using the 'getDigest()' method.</i>	
Crypto::encrypt_blowfish()	34
<i>Encrypts the specified data using Blowfish</i>	
Crypto::getSupportedHashAlgorithms()	35
<i>Returns an array of supported hashing algorithms on the current PHP enabled web server.</i>	
Crypto::hash()	35
<i>Returns the hashed \$data parameter according to the defined \$algorithm parameter.</i>	
Crypto::sha1()	36
<i>Returns an SHA1 hashed string.</i>	
Crypto::setAlgorithm()	36
<i>Sets the algorithm for the Crypto component for use with getDigest.</i>	
Crypto::md5()	35
<i>Returns a hashed MD5 string.</i>	
Crypto::encrypt_aes256()	33
<i>Encrypts the specified data using AES 256 encryption</i>	
Crypto::encrypt_3des()	33
<i>Encrypts the specified data using Triple DES.</i>	
constructor Crypto::construct()	31
<i>Initializes the Crypto component with the hasing algorithm defined in agilephp.xml</i>	
Crypto	31
<i>AgilePHP :: Crypto</i>	
<i>Provides one way hashing and encryption</i>	
constructor AgilePHP_RemotingException::construct()	27
<i>Deliver remoting exceptions in JSON format and halt execution.</i>	
constructor AgilePHP_Exception::construct()	25
Crypto::crc32()	31
<i>Returns an CRC32 hashed string.</i>	
Crypto::createIV()	32
Crypto::decrypt_blowfish()	32

<u>Decrypts a string previously encrypted with encrypt_blowfish</u>	32
<u>Crypto::decrypt_aes256()</u>	32
<i>Decrypts the specified data which was previously encrypted using AES 256</i>	
<u>Crypto::decrypt_3des()</u>	32
<i>Decrypts Triple DES data</i>	
<u>Crypto::sha256()</u>	37
<i>Returns an SHA256 hashed string.</i>	
<u>Crypto::sha384()</u>	37
<i>Returns an SHA384 hashed string.</i>	
<u>constructor InvocationContext:: __construct()</u>	146
<i>Initializes the InvocationContext.</i>	
<u>constructor InterceptorProxy:: __construct()</u>	143
<i>Initializes the intercepted class by creating a new instance, passing in any constructor arguments as required. Class level interceptors are invoked here, as well as dependancy injections via #@In.</i>	
<u>constructor User::User()</u>	134
<u>constructor Role:: __construct()</u>	132
<u>constructor BaseController:: __construct()</u>	168
<i>Creates a new instance of default renderer</i>	
<u>constructor BaseModelActionController:: __construct()</u>	171
<u>constructor BaseModelXslController:: __construct()</u>	186
<u>constructor BaseModelXmlController:: __construct()</u>	183
<u>constructor BaseModelController:: __construct()</u>	174
<i>Initializes parent and executes a default SELECT query for the first 25 records.</i>	
<u>constructor AnnotatedProperty:: __construct()</u>	104
<u>constructor AnnotatedMethod:: __construct()</u>	101
<u>constructor Interception:: __construct()</u>	55
<i>Initializes the Interception</i>	
<u>constructor Form:: __construct()</u>	38
<u>Crypto::sha512()</u>	37
<i>Returns an SHA512 hashed string.</i>	
<u>constructor PersistenceManager:: __construct()</u>	64
<i>Initializes the PersistenceManager object based on persistence.xml configuration. Each of the databases, tables, and columns are transformed into corresponding AgilePHP persistence objects.</i>	
<u>constructor Remoting:: __construct()</u>	82
<u>constructor AnnotatedClass:: __construct()</u>	98
<i>Initializes the AnnotatedClass by parsing the class file for annotations.</i>	
<u>constructor Validator:: __construct()</u>	90
<u>constructor Upload:: __construct()</u>	88
<u>Crypto.php</u>	5
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	

D

<u>Database::setId()</u>	228
<i>Sets the database id. This is used internally by AgilePHP to distinguish between multiple data sources configured in persistence.xml</i>	
<u>Database::setHostname()</u>	228
<i>Sets the hostname of the database server</i>	
<u>Database::hasTables()</u>	228

<i>Returns boolean indicator based on the presence of any Table instances configured for the database.</i>	
Database::setName()	229
<i>Sets the name of the database</i>	
Database::setPassword()	229
<i>Sets the password which is used to connect to the database</i>	
Database::setUsername()	230
<i>Sets the username required to access the database</i>	
Database::setType()	230
<i>Sets the type of database server</i>	
Database::setTables()	229
<i>Sets the array of Table instances which represent a table in the physical database.</i>	
Database::getUsername()	227
<i>Returns the username which is used to connect to the database</i>	
Database::getType()	227
<i>Returns the database server type.</i>	
Database::getHostname()	226
<i>Returns the hostname of the database server.</i>	
Database::addTable()	226
<i>Pushes a new Table instance onto the stack</i>	
Database	225
<i>AgilePHP :: Database</i>	
<i>Represents a database in the AgilePHP persistence component.</i>	
Database::getId()	226
<i>Returns the unique database identifier.</i>	
Database::getName()	227
<i>Returns the name of the database</i>	
Database::getTables()	227
<i>Returns an array of Table instances which represent a table in the physical database.</i>	
Database::getPassword()	227
<i>Returns the password thats used to connect to the database</i>	
Database.php	198
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	

F

ForeignKey::getReferencedTable()	233
<i>Returns the physical name of the referenced table.</i>	
ForeignKey::getReferencedTableInstance()	233
<i>Returns an instance of the referenced Table.</i>	
ForeignKey::getSelect()	233
<i>Gets the name of the foreign table's column which should be displayed in a drop-down in place of the actual foreign key value.</i>	
ForeignKey::getSelectedColumnInstance()	234
<i>Returns the foreign (parent) column instance which contains a select="true" configuration in persistence.xml. If a column is not explicitly set, the referenced column instance is returned.</i>	
ForeignKey::getReferencedController()	233
<i>Returns the name of the referenced MVC controller responsible</i>	

for the management of the referenced domain model.	
ForeignKey::getReferencedColumnInstance()	233
Returns an instance of the referenced column	
ForeignKey::getName()	232
Returns the name of the foreign key	
ForeignKey::getOnDelete()	232
Returns the configured SQL 'OnDelete' action for the foreign key.	
ForeignKey::getOnUpdate()	232
Returns the configured SQL 'OnUpdate' action for the foreign key.	
ForeignKey::getReferencedColumn()	232
Returns the name of the referenced (parent) column.	
ForeignKey::getType()	234
Returns the data type being stored in the foreign key column.	
ForeignKey::setFkColumn()	234
The name of the column which stores this foreign key	
ForeignKey::setReferencedController()	236
Sets the name of the referenced MVC controller responsible for the management of the referenced domain model.	
ForeignKey::setReferencedTable()	237
Sets the name of the referenced table.	
ForeignKey::setSelect()	237
Sets the name of the foreign table's column which should be displayed in a drop-down in place of the actual foreign key value.	
ForeignKey::setType()	237
Sets the foreign key value data type.	
ForeignKey::setReferencedColumn()	236
Sets the name of the referenced (parent) column.	
ForeignKey::setOnUpdate()	236
Sets the SQL 'OnUpdate' action for the foreign key. This action is performed automatically by the database server anytime the primary key in the foreign (parent) table is updated.	
ForeignKey::setFkTable()	235
Sets the physical name of the foreign key table.	
ForeignKey::setName()	235
Sets the name of the foreign key. This is the actual name given to the foreign key according to the database server.	
ForeignKey::setOnDelete()	235
Sets the SQL 'OnDelete' action which is invoked when the foreign (parent) key is updated.	
ForeignKey::getFkTable()	232
Returns the name of the table that stores this foreign key.	
ForeignKey::getFkColumn()	231
Returns the physical name of the column which stores the foreign key.	
Form::getMode()	40
Returns the form presentation mode.	
Form::getModel()	40
Returns the domain model instance being used to create the form.	
Form::getName()	40
Returns the form's HTML name attribute value	
Form::getRequestToken()	40
Returns the anti-CSFR token	
Form::getId()	39
Returns the form's id attribute value	

Form::getHTML()	39
<i>Returns the form in HTML format</i>	
Form	38
<i>AgilePHP :: Form</i>	
<i>Generates a form by reverse engineering the specified domain model.</i>	
Form::getAction()	38
<i>Returns the forms action attribute value</i>	
Form::getEnctype()	39
<i>Returns the form's enctype attribute value</i>	
Form::getForeignKeySelection()	39
<i>Returns an HTML select drop-down which contains a list of values for foreign key columns used in one-to-many relationships.</i>	
Form::getXSL()	40
<i>Returns the form in XSL format</i>	
Form::setAction()	41
<i>Sets the form action attribute value</i>	
Form::setRequestToken()	43
<i>Sets the anti-CSFR token to render within the form. Note that the CSFR token should be different on every form!</i>	
ForeignKey.php	199
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
ForeignKey	230
<i>AgilePHP :: ForeignKey</i>	
<i>Represents a foreign key in the AgilePHP persistence component.</i>	
ForeignKey::getColumnInstance()	231
<i>Returns the column instance where the foreign key resides</i>	
Form::setName()	43
<i>Sets the HTML name attribute value</i>	
Form::setModel()	42
<i>Sets the domain model that is used to create the form</i>	
Form::setEnctype()	41
<i>Sets the form's enctype attribute value</i>	
Form::setId()	42
<i>Sets the HTML id attribute value.</i>	
Form::setMode()	42
<i>Sets the form presentation mode.</i>	
Form.php	6
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	

IdentityManager::setUsername()	128
<i>Sets the username for the identity which IdentityManager is managing.</i>	
IdentityModel	129
<i>AgilePHP :: IdentityModel</i>	
<i>Interface for Identity domain model</i>	
IdentityManager::setRole()	128
<i>Sets the identity's role</i>	
IdentityManager::setPasswordResetUrl()	127

<i>Sets the url used to reset the identity's password.</i>	
<u>IdentityManager::setModel()</u>	127
<i>Sets the domain object model which the IdentityManager will manage.</i>	
<u>IdentityManager::setPassword()</u>	127
<i>Sets the password for the identity which IdentityManager is managing.</i>	
<u>IdentityModel::getCreated()</u>	129
<u>IdentityModel::getEmail()</u>	129
<u>IdentityModel::getSessionId()</u>	130
<u>IdentityModel::getUsername()</u>	130
<u>IdentityModel::getRole()</u>	130
<u>IdentityModel::getPassword()</u>	129
<u>IdentityModel::getLastLogin()</u>	129
<u>IdentityManager::setLastLogin()</u>	126
<i>Sets the dateTime when this identity last logged in</i>	
<u>IdentityManager::setEmail()</u>	126
<i>Sets the identity's email address.</i>	
<u>IdentityManager::isLoggedIn()</u>	124
<i>Returns boolean response based on the identity's logged in status.</i>	
<u>IdentityManager::login()</u>	124
<i>Authenticates/logs in an identity and returns a boolean response.</i>	
<u>IdentityManager::hasRole()</u>	123
<i>Checks to see if the identity has the specified role.</i>	
<u>IdentityManager::getUsername()</u>	123
<i>Returns the username for the identity which IdentityManager is managing.</i>	
<u>IdentityManager::getPasswordResetUrl()</u>	123
<i>Returns the url which is sent to the identity that when clicked resets their password.</i>	
<u>IdentityManager::getRole()</u>	123
<i>Returns the Role object</i>	
<u>IdentityManager::logout()</u>	124
<i>Destroys the current SessionScope.</i>	
<u>IdentityManager::merge()</u>	124
<i>Updates the identity using the current state of the domain model object which the IdentityManager is managing.</i>	
<u>IdentityManager::revokeRole()</u>	126
<i>Revokes/removes a role from the identity.</i>	
<u>IdentityManager::setCreated()</u>	126
<i>Denotes when the identity was created.</i>	
<u>IdentityManager::resetPassword()</u>	125
<i>Resets the password to a hashed random string. This operation uses the AgilePHP Crypto component to ensure standard hashing across the application.</i>	
<u>IdentityManager::refresh()</u>	125
<i>Refreshes an entity by performing a 'find' operation on the domain object model which the IdentityManager is managing.</i>	
<u>IdentityManager::persist()</u>	125
<i>Persists a new identity.</i>	
<u>IdentityModel::setCreated()</u>	130
<u>IdentityModel::setEmail()</u>	130
<u>InvocationContext::getMethod()</u>	147
<i>Returns the name of the method which was intercepted.</i>	
<u>InvocationContext::getParameters()</u>	147
<i>Returns the parameters being passed into the intercepted method.</i>	
<u>InvocationContext::getInterceptor()</u>	147
<i>Returns the instance of the interceptor annotation which caused the interception.</i>	

InvocationContext::\$proceed	146
InterceptorProxy::__unset()	145
<i>Magic PHP unset function.</i>	
InvocationContext	146
<i>AgilePHP :: InvocationContext</i>	
<i>Provides environment and method execution details t interceptors.</i>	
InvocationContext::getTarget()	147
<i>Returns the target instance which was intercepted.</i>	
InvocationContext::proceed()	148
<i>Causes the intercepted invocation to proceed, using the method and parameters which are present in the current InvocationContext.</i>	
In	153
<i>AgilePHP :: In</i>	
<i>Dependency injection @In interceptor</i>	
In::\$class	153
In.php	151
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
InvocationContext::setParameters()	148
<i>Changes the method parameters. This is used by the interceptor to cause a different set of parameters to be passed into the method which gets invoked.</i>	
InvocationContext::setMethod()	148
<i>Changes the name of the method. This is used by the interceptor to cause a different method to be executed than initially requested.</i>	
InterceptorProxy::__set()	145
<i>Magic PHP property mutator</i>	
InterceptorProxy::__isset()	145
<i>Magic PHP isset. Checks a property to see if its set</i>	
IdentityModel::setSessionId()	131
IdentityModel::setUsername()	132
IdentityModel::setRole()	131
IdentityModel::setPassword()	131
IdentityModel::setLastLogin()	131
InterceptorProxy.php	141
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
InvocationContext.php	142
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
InterceptorProxy::__call()	144
<i>Magic PHP method executor.</i>	
InterceptorProxy::__get()	144
<i>Magic PHP property accessor</i>	
InterceptorProxy::getInterceptedInstance()	144
<i>Returns the instance of the intercepted class.</i>	
InterceptorProxy::getInstance()	143
<i>Returns a singleton instance of the intercepted class</i>	
InterceptorProxy	143
<i>AgilePHP :: InterceptorProxy</i>	
<i>Dynamic proxy template for intercepted classes</i>	
IdentityManager::getPassword()	122
<i>Returns the password for the identity which IdentityManager is managing. Unless</i>	

overwritten, this is the persisted value.	122
IdentityManager::getModel()	122
Returns the domain object model which the IdentityManager is managing.	
Identity::getRole()	48
(non-PHPdoc)	
Identity::getUsername()	48
(non-PHPdoc)	
Identity::getPasswordResetUrl()	48
(non-PHPdoc)	
Identity::getPassword()	47
(non-PHPdoc)	
Identity::getLastLogin()	47
(non-PHPdoc)	
Identity::getModel()	47
(non-PHPdoc)	
Identity::hasRole()	48
(non-PHPdoc)	
Identity::isLoggedIn()	49
(non-PHPdoc)	
Identity::persist()	50
(non-PHPdoc)	
Identity::refresh()	50
Refreshes the IdentityManager domain object model's state by performing a fresh sql SELECT query.	
Identity::merge()	50
(non-PHPdoc)	
Identity::logout()	49
(non-PHPdoc)	
Identity::login()	49
Authenticates the specified username and password.	
Identity::getInstance()	47
Returns a singleton instance of the 'Identity' component.	
Identity::getEmail()	47
(non-PHPdoc)	
i18n::getInstance()	44
Returns a singleton instance of i18n	
i18n::getLocale()	44
Returns the language locale being used to translate	
i18n::getDomain()	44
Returns the messaging domain.	
i18n	43
AgilePHP :: i18n (internationalization)	
Identity.php	8
AgilePHP Framework :: The Rapid "for developers" PHP5 framework Copyright (C) 2009 Make A Byte, inc	
Interception.php	9
AgilePHP Framework :: The Rapid "for developers" PHP5 framework Copyright (C) 2009 Make A Byte, inc	
i18n::setDomain()	44
Sets the messaging domain. This is the name of your .PO/.MO files.	
i18n::setLocale()	45
Sets the language locale.	
Identity::forgotPassword()	46

(non-PHPdoc)	46
Identity::getCreated()	46
(non-PHPdoc)	
Identity::delete()	46
(non-PHPdoc)	
Identity	46
AgilePHP :: Identity	
i18n::translate()	45
Performs language translation based on the configured locale.	
Identity::resetPassword()	50
(non-PHPdoc)	
Identity::revokeRole()	51
(non-PHPdoc)	
Interceptor	114
AgilePHP :: Interceptor	
Annotation used to create an interceptor	
IdentityManager.php	117
AgilePHP Framework :: The Rapid "for developers" PHP5 framework	
Copyright (C) 2009 Make A Byte, inc	
Interceptor.php	112
AgilePHP Framework :: The Rapid "for developers" PHP5 framework	
Copyright (C) 2009 Make A Byte, inc	
Interception::getSourceCode()	56
Returns the PHP file content to be parsed.	
Interception::getMethod()	56
Returns the name of the target method to intercept	
Interception::getProperty()	56
Returns the name of the property to intercept	
IdentityModel.php	118
AgilePHP Framework :: The Rapid "for developers" PHP5 framework	
Copyright (C) 2009 Make A Byte, inc	
IdentityManager	121
AgilePHP :: IdentityManager	
Interface for Identity component	
IdentityManager::getEmail()	122
Returns the identity's email address	
IdentityManager::getLastLogin()	122
Returns the dateTime the identity last logged in	
IdentityManager::getCreated()	122
Returns the dateTime when the identity was created.	
IdentityManager::forgotPassword()	121
Sends the identity an email to the address stored in the stateful domain object model being managed by IdentityManager. Uses the state of the 'email' field within the domain object model the IdentityManager is managing.	
IdentityManager::delete()	121
Deletes the domain object model which the IdentityManager is managing.	
Interception::getInterceptor()	56
Returns the interceptor instance which handles the intercepted call.	
Interception::getClass()	55
Returns the name of the target class being intercepted	
Identity::setModel()	52
(non-PHPdoc)	

Identity::setPassword()	52
<i>Sets encrypted/hashed password according to the configuration of the 'Crypto' component.</i>	
Identity::setLastLogin()	52
<i>(non-PHPdoc)</i>	
Identity::setEmail()	51
<i>(non-PHPdoc)</i>	
Identity::setCreated()	51
<i>(non-PHPdoc)</i>	
Identity::setPasswordResetUrl()	53
<i>(non-PHPdoc)</i>	
Identity::setRole()	53
<i>(non-PHPdoc)</i>	
Interception::createInterceptedTarget()	55
<i>Creates a new intercepted target instance. The target is created by modifying the source code of the class being intercepted to *classname*_Intercepted.</i>	
Interception::createInterceptorProxy()	55
<i>Creates and loads a dynamic proxy class which performs interceptions on the class created by Interception::createInterceptedTarget().</i>	
Interception	54
<i>AgilePHP :: Interception</i>	
<i>Performs interceptions by creating a dynamic proxy for intercepted classes. The proxy invokes the intended calls after inspecting (and/or intercepting) it according to the annotations in the intercepted object.</i>	
Identity::destruct()	54
<i>Destructor</i>	
Identity::setUsername()	54
<i>(non-PHPdoc)</i>	
i18n.php	7
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	

L

Logger::info()	58
<i>Writes an 'info' log level entry.</i>	
Logger::setConfig()	58
<i>Used by AgilePHP main framework class to automatically load Logger based on agilephp.xml configuration.</i>	
Logger::warn()	59
<i>Writes a 'warn' log level entry.</i>	
Logger::getInstance()	58
<i>Factory method that creates an instance of the configured Logger. If a logger has not been configured in agilephp.xml, an instance of 'FileLogger' is returned.</i>	
Logger::error()	57
<i>Writes an 'error' log level entry.</i>	
Logger	57
<i>AgilePHP :: Logger</i>	
<i>Performs basic logging functionality</i>	
Logger::debug()	57
<i>Writes a 'debug' log level entry.</i>	

Logger.php	10
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	

M

MVC::setDefaultController()	63
<i>Sets the name of the default controller which is used if one is not specified in the request URI. Default is 'IndexController'.</i>	
MVC::setDefaultAction()	62
<i>Sets the name of the default action method if one is not specified in the request URI. Default is 'index'.</i>	
MVC::setConfig()	62
<i>Initializes the MVC component with agilephp.xml configuration.</i>	
MVC::processRequest()	62
<i>Parses the current request URI to obtain the controller, action method, and arguments present for this request and then performs the invocation. If these parameters ARE NOT present, the default controller and default action method are used instead.</i>	
MVC::setDefaultRenderer()	63
<i>Sets the name of the default view renderer. Default is 'PHTMLRenderer'.</i>	
MySQLDialect.php	250
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
MySQLDialect::drop()	254
<i>Drop/destroy the database.</i>	
MySQLDialect::create()	253
<i>Creates the database as defined in persistence.xml</i>	
MySQLDialect	253
<i>AgilePHP :: MySQLDialect</i>	
<i>Handles MySQL specific queries</i>	
MVC::getInstance()	61
<i>Returns a singleton instance of MVC</i>	
MVC::getDefaultRenderer()	61
<i>Returns the name of the default view renderer</i>	
MVC::createDefaultRenderer()	60
<i>Returns a new instance of the default view renderer</i>	
MVC::createCustomRenderer()	59
<i>Returns a new instance of the specified renderer. The renderer is loaded from the web app 'classes' directory.</i>	
MVC	59
<i>AgilePHP :: Model-View-Control (MVC)</i>	
MVC::createRenderer()	60
<i>Returns a new instance of the specified view renderer</i>	
MVC::getAction()	60
<i>Returns the action currently being invoked.</i>	
MVC::getDefaultController()	61
<i>Returns the name of a default controller if one is not specified in the request URI. Default is 'IndexController'.</i>	
MVC::getDefaultAction()	61
<i>Returns the name of a default action method if one is not specified in the request URI. Default is 'index'.</i>	
MVC::getController()	61

Returns the name of the controller currently in use.

MVC.php	11
AgilePHP Framework :: The Rapid "for developers" PHP5 framework	
Copyright (C) 2009 Make A Byte, inc	

P

PersistenceManager::quote()	76
<i>Quotes a string so its theoretically safe to pass into a statement without the worry of SQL injection.</i>	
PersistenceManager::query()	76
<i>Executes a raw SQL query using PDO::query</i>	
PersistenceManager::previousExists()	76
<i>Returns boolean result based on whether or not a 'previous page' result is available during a pagination request.</i>	
PersistenceManager::rollBack()	77
<i>Rolls back a transaction.</i>	
PersistenceManager::setDistinct()	77
<i>Adds an SQL distinct clause to 'find' operation.</i>	
PersistenceManager::setMaxResults()	78
<i>Sets the maximum number of records to return in a result list</i>	
PersistenceManager::setGroupBy()	77
<i>Sets the SQL 'group by' clause.</i>	
PersistenceManager::prepare()	75
<i>Prepares an SQL prepared statement</i>	
PersistenceManager::persist()	75
<i>Persists a domain model object</i>	
PersistenceManager::getTableByModelName()	73
<i>Returns a 'Table' object representing the table configured in persistence.xml as the AgilePHP 'Identity' table.</i>	
PersistenceManager::getTableByModel()	73
<i>Returns the 'Table' object which is mapped to the specified \$model.</i>	
PersistenceManager::getTableByName()	74
<i>Returns a 'Table' object by its name as configured in persistence.xml</i>	
PersistenceManager::getTableName()	74
<i>Returns the table name for the model defined in the extension class</i>	
PersistenceManager::nextExists()	75
<i>Returns boolean result based on whether or not a 'next page' result is available during a pagination request.</i>	
PersistenceManager::merge()	74
<i>Merges/updates a persisted domain model object</i>	
PersistenceManager::setModel()	78
<i>Sets the domain model the PersistenceManager is to manipulate.</i>	
PersistenceManager::setOrderBy()	78
<i>Sets the SQL 'order by' clause.</i>	
PHTMLRenderer::renderComponent()	190
<i>Renders a component view by dumping all 'store' variables to locally scoped (page) variables. The view is expected to be in <webapp>/component/<component_name>/view.</i>	
PHTMLRenderer::render()	189
<i>Renders a view by dumping all 'store' variables to locally scoped (page) variables. The view is expected to be in <webapp>/view.</i>	

PHTMLRenderer	189
<i>AgilePHP :: MVC PHTMLRenderer</i>	
<i>Provides base rendering implementation for PHTML (PHP) views.</i>	
PersistenceRenderer.php	200
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
PersistenceRenderer	238
<i>AgilePHP :: PersistenceRenderer</i>	
<i>Handles rendering persisted data to HTML</i>	
PersistenceRenderer::toFormElement()	239
<i>Creates an HTML input form element for the specified column.</i>	
PersistenceRenderer::isBit()	238
<i>Returns boolean response based on the configured 'type' attribute for the specified column.</i>	
PHTMLRenderer.php	164
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
PersistenceManager::truncate()	81
<i>Truncates the table for the specified domain model object</i>	
PersistenceManager::setRestrictions()	79
<i>Sets WHERE clause restrictions</i>	
PersistenceManager::setPage()	79
<i>Sets the pagination page number and performs an SQL query to populate the 'resultList' and 'resultCount' properties with their appropriate values for the specified page.</i>	
PersistenceManager::setRestrictionsLogicOperator()	80
<i>Sets the restriction operator (and/or) used in SQL WHERE clause.</i>	
PersistenceManager::toAccessor()	80
<i>Creates an accessor method from the \$property parameter. The \$property will be returned with the prefix 'get' and the first letter of the property uppercased.</i>	
PersistenceManager::toMutator()	81
<i>Creates a mutator method from the \$property parameter. The \$property will be returned with the prefix 'set' and the first letter of the property uppercased.</i>	
PersistenceManager::toBigInt()	80
<i>Returns the specified number as a 'bigint' 64-bit number.</i>	
PersistenceManager::getSessionTable()	73
<i>Returns a SimpleXMLElement representing the table configured in persistence.xml as an AgilePHP 'SessionScope' session table.</i>	
PersistenceManager::getSessionModel()	73
<i>Returns the domain model object responsible for 'SessionScope' sessions.</i>	
PersistenceManager::execute()	67
<i>Executes a prepared statement (with parameters)</i>	
PersistenceManager::exec()	67
<i>Executes an SQL statement and returns the number of rows affected by the query.</i>	
PersistenceManager::drop()	66
<i>Drops the database specified in persistence.xml</i>	
PersistenceManager::executeCountQuery()	67
<i>Executes an SQL count query for total number of records in the database.</i>	
PersistenceManager::executeQuery()	68
<i>Executes the current SQL statement as defined in 'sql'.</i>	
PersistenceManager::getColumnNameByProperty()	68
<i>Returns the column 'name' attribute value configured in persistence.xml for the specified column 'property' attribute.</i>	

PersistenceManager::find()	68
<i>Attempts to locate the specified model by primary key value.</i>	
PersistenceManager::delete()	66
<i>Deletes a persisted domain model object</i>	
PersistenceManager::createQuery()	66
<i>Sets the SQL statement to use when calling executeQuery.</i>	
PersistenceManager::beginTransaction()	64
<i>Begins a transaction</i>	
PersistenceManager	64
<i>AgilePHP :: PersistenceManager</i>	
<i>Facade for working with persistence operations</i>	
PersistenceManager::commit()	65
<i>Commits an already started transaction.</i>	
PersistenceManager::compareModels()	65
<i>Checks the two defined model parameters to see if they are equal. The class name, property name, type and value.</i>	
PersistenceManager::create()	66
<i>Creates the database context as defined in persistence.xml.</i>	
PersistenceManager::connect()	65
<i>Establishes a connection to the specified database.</i>	
PersistenceManager::getCount()	69
<i>Returns the total number of records in the database table</i>	
PersistenceManager::getDistinct()	69
<i>Returns the 'distinct' column to use in an SQL SELECT statement if one has been defined.</i>	
PersistenceManager::getPropertyNameByColumn()	71
<i>Returns the 'property' name configured in persistence.xml for the specified column 'name' attribute.</i>	
PersistenceManager::getPreviousResultList()	71
<i>Sets the result list up with a 'previous page' of a pagination request</i>	
PersistenceManager::getPageCount()	71
<i>Returns the total number of pages. This is calculated by dividing the total number of records by 'maxResults'.</i>	
PersistenceManager::getQuery()	72
<i>Returns the current SQL query</i>	
PersistenceManager::getResultCount()	72
<i>Returns the total number of records in the current result list.</i>	
PersistenceManager::getSelectedDatabase()	72
<i>Returns the database currently being used by the PersistenceManager.</i>	
PersistenceManager::getResultList()	72
<i>Returns a result set from the database. The result list is returned as an array of stdClass objects, each representing a row in the database.</i>	
PersistenceManager::getPage()	71
<i>Returns the current page number</i>	
PersistenceManager::getOrderBy()	70
<i>Returns an associative array containing the current 'orderBy' clause. The results are returned with the name of the column as the index and the direction as the value.</i>	
PersistenceManager::getIdentityTable()	69
<i>Returns a SimpleXMLElement representing the table configured in persistence.xml as an AgilePHP 'Identity' table.</i>	
PersistenceManager::getIdentityModel()	69
<i>Returns the domain object model responsible for 'Identity' persistence.</i>	
PersistenceManager::getMaxResults()	70

<i>Returns the maximum number of results to retrieve. This translates to an SQL LIMIT clause during SELECT operations.</i>	
PersistenceManager::getModel()	70
<i>Returns the domain model which the PersistenceManager is manipulating.</i>	
PersistenceManager::getNextResultList()	70
<i>Sets the result list up with a 'next page' of a pagination request</i>	
PersistenceManager::getModelName()	70
<i>Returns the model's class name.</i>	
PersistenceManager.php	12
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	

R

Restrict::\$roles	154
Restrict::requireRole()	154
RequestScope.php	259
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
Restrict::\$role	154
Restrict::\$message	154
Role::setName()	133
<i>Sets the name of the role</i>	
Restrict.php	152
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
Restrict	153
<i>AgilePHP :: Restrict</i>	
<i>Interceptor for @Restrict annotations</i>	
RequestScope	264
<i>AgilePHP :: RequestScope</i>	
<i>Stores PHP \$_POST variables, performs sanitizing, and has built in support to guard against CSFR attacks using a double cookie submit approach.</i>	
RequestScope::createToken()	265
<i>Creates a 20-30 character token used to guard against CSFR attacks.</i>	
RequestScope::sanitize()	266
<i>Sanitizes the specified data by running it through htmlspecialchars, addslashes, and strip_tags.</i>	
RequestScope::set()	267
<i>Sets a store variable.</i>	
README	283
RequestScope::invalidate()	266
<i>Expires request token cookie.</i>	
RequestScope::getSanitized()	266
<i>Returns a sanitized store variable which protects against SQL injection, XSS and XSFR attacks. The variable value is passed first through strip_tags, followed by addslashes and finally htmlspecialchars. Use htmlspecialchars_decode to decode the sanitized value.</i>	
RequestScope::get()	265
<i>Returns a store variable.</i>	
RequestScope::getInstance()	265

<i>Returns singleton instance of RequestScope</i>	
RequestScope::getParameters()	265
<i>Returns the variable store.</i>	
Role::setDescription()	133
<i>Sets the description of the role.</i>	
Role::getName()	133
<i>Returns the name of the role</i>	
Remoting::getBaseJS()	84
<i>Returns the raw JavaScript contents of the AgilePHP.js file and pre-configures the library with a default AgilePHP.debug and AgilePHP.Remoting.controller value.</i>	
Remoting::getClass()	84
<i>Returns the name of the class being remoted</i>	
Remoting::getSessionId()	84
<i>Returns the current session id. If a session is not active a new session is created and the id is returned.</i>	
Remoting::destroySession()	83
<i>Destroys the session used for stateful remoting</i>	
Remoting::createStub()	83
<i>Creates a dynamic javascript proxy stub/interface used for remoting PHP classes. This method handles both standard and intercepted classes.</i>	
Remoting	81
<i>AgilePHP :: Remoting</i>	
<i>Provides client side JavaScript remoting to PHP objects. Handles marshalling/unmarshalling of JSON objects between the client and server.</i>	
Remoting::createInterceptedStub()	82
<i>Creates a dynamic javascript proxy stub/interface used for remoting AgilePHP intercepted classes</i>	
Remoting::createStandardStub()	82
<i>Creates a dynamic javascript proxy stub/interface used for remoting standard PHP classes</i>	
Remoting::invoke()	84
<i>Invokes a non-persistent/stateful instance of the requested class/method passing in arguments if any were defined. This non-stateful approach is how most RPC web services work.</i>	
Remoting::invokeIntercepted()	85
<i>Performs RMI invocation on an intercepted class (non-stateful).</i>	
Role.php	119
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
Role	132
<i>AgilePHP :: Role</i>	
<i>Role associated with an Identity</i>	
Role::getDescription()	133
<i>Returns the role description</i>	
RemoteMethod	115
<i>AgilePHP :: RemoteMethod</i>	
<i>Annotation which authorizes a method for Remoting</i>	
RemoteMethod.php	113
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
Remoting::invokeInterceptedStateful()	85
<i>Invokes the specified intercepted class/method/args using a stateful instance stored in SessionScope. If a stateful instance does not exist, a new instance is created and stored in the SessionScope</i>	

for future calls.	
Remoting::invokeStateful()	85
<i>Invokes the specified class/method/args using a stateful instance stored in SessionScope. If a stateful instance does not exist, a new instance is created and stored in the SessionScope for future calls.</i>	
Remoting::setClass()	85
<i>Sets the name of the class to remote</i>	
Remoting.php	13
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	

S

SessionScope	269
<i>AgilePHP :: SessionScope</i>	
SessionScope::clear()	270
<i>Clears the current Session.</i>	
SessionScope::destroy()	270
<i>Clears the SessionScope store and deletes the Session ActiveRecord from the database.</i>	
Session::setId()	269
<i>Sets the session id</i>	
Session::setData()	268
<i>Stores the serialized session</i>	
Session::getData()	268
<i>Returns the serialized session data</i>	
Session::getId()	268
<i>Returns the session id</i>	
Session::setCreated()	268
<i>Timestamp indicating when the session was created</i>	
SessionScope::get()	270
<i>Returns the value corresponding to the specified key stored in the current Session.</i>	
SessionScope::getInstance()	270
<i>Returns a singleton instance of SessionScope</i>	
SessionScope::set()	271
<i>Sets a new Session variable.</i>	
SessionScope::setSessionId()	272
<i>Sets the session id and attempts to restore a previously persisted Session.</i>	
SessionScope::destruct()	272
<i>Persist the Session data state to database just before the object is destroyed.</i>	
SessionScope::persist()	271
<i>Persists a serialized instance of the current Session.</i>	
SessionScope::isPersisted()	271
<i>Returns boolean flag indicating whether or not the current session data is persisted.</i>	
SessionScope::getSession()	271
<i>Returns the session domain model object which maintains the id and data for the current Session ActiveRecord.</i>	
SessionScope::getSessionId()	271
<i>Returns the session id for the current Session.</i>	

Session::getCreated()	267
<i>Returns the timestamp when the session was created</i>	
Session	267
<i>AgilePHP :: Session</i>	
<i>Session domain model object.</i>	
Scope::getSessionScope()	87
<i>Returns a singleton instance of RequestScope.</i>	
SQLDialect.php	251
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
SQLiteDialect.php	252
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
Scope::getRequestScope()	87
<i>Returns a singleton instance of RequestScope.</i>	
Scope::getInstance()	87
<i>Returns a singleton instance of Scope.</i>	
Scope	86
<i>AgilePHP :: Scope</i>	
<i>Facade for AgilePHP scopes.</i>	
Scope::getApplicationScope()	86
<i>Returns a singleton instance of ApplicationScope.</i>	
SQLDialect	254
<i>AgilePHP :: SQLDialect</i>	
<i>Defines contract for dealing with vendor specific SQL queries</i>	
SQLDialect::create()	254
SQLiteDialect::truncate()	256
Session.php	260
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
SessionScope.php	261
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	
SQLiteDialect::drop()	256
<i>Delete the database.</i>	
SQLiteDialect::create()	255
<i>CREATE TABLE "phpunit2" ("id" INTEGER NOT NULL ,</i>	
<i>"fld1" VARCHAR NOT NULL , "fld2" VARCHAR DEFAULT 'This is</i>	
<i>nullable', "fld3" VARCHAR NOT NULL , "fld4" TEXT NOT NULL ,</i>	
<i>PRIMARY KEY ("id", "fld1"))</i>	
SQLDialect::drop()	255
SQLiteDialect	255
<i>AgilePHP :: SQLiteDialect</i>	
<i>Responsible for SQLite specific database operations</i>	
Scope.php	14
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	

T

Table::isIdentity()	244
<i>Returns true if the table is configured as an AgilePHP Identity table, false otherwise.</i>	

<u>Table::isSession()</u>	245
<i>Returns boolean indicator based on whether or not this is a table responsible for AgilePHP session table.</i>	
<u>Table::isVisible()</u>	245
<i>Returns boolean response based on the 'visible' attribute in persistence.xml for the column which contains the specified \$property.</i>	
<u>Table::hasForeignKeyReferences()</u>	244
<i>Returns boolean response based on whether or not there are any other tables which have foreign keys referencing this table.</i>	
<u>Table::hasForeignKey()</u>	244
<i>Returns boolean response based on the presence of a column which is configured in persistence.xml as a foreign key.</i>	
<u>Table::hasBlobColumn()</u>	243
<i>Returns boolean response based on the presence of a column which is configured in persistence.xml as a blob data type.</i>	
<u>Table::hasColumns()</u>	244
<i>Returns boolean response based on the presence of one or more Column instances.</i>	
<u>Table::setColumns()</u>	245
<i>Sets the array of columns for the table</i>	
<u>Table::setDescription()</u>	246
<i>Sets the description field for the table. This field is used by visual elements which reference the tables description.</i>	
<u>Table::setName()</u>	247
<i>Sets the table name</i>	
<u>Table::setValidate()</u>	248
<i>Boolean field used to toggle automatic data validation for the table</i>	
<u>Table::setModel()</u>	247
<i>Sets the name of the domain model responsible for the table</i>	
<u>Table::setIsSession()</u>	247
<i>Sets the boolean field used to identity an AgilePHP Session table.</i>	
<u>Table::setDisplay()</u>	246
<i>Text field used to configure a 'display name'. If a value is specified this is the name that is used in visual elements such as forms and result lists when referencing the table.</i>	
<u>Table::setIsIdentity()</u>	246
<i>Sets the boolean field used to identify an AgilePHP Identity table.</i>	
<u>Table::getViewDisplayName()</u>	243
<i>Returns a name suitable for display to end users. If a friendly name has been configured for this table (by providing a display attribute value in persistence.xml for the Column), then this value is returned, otherwise the table name attribute value is returned instead.</i>	
<u>Table::getValidate()</u>	243
<i>Returns the boolean field used to toggle automatic data validation.</i>	
<u>Table::getDescription()</u>	241
<i>Returns the table description</i>	
<u>Table::getDisplay()</u>	241
<i>Returns the friendly display name for the table.</i>	
<u>Table::getColumns()</u>	240
<i>Returns an array of Column instances which are configured for the table.</i>	
<u>Table::getColumnNameByProperty()</u>	240
<i>Looks up a 'Column' name by its persistence.xml 'property' attribute.</i>	
<u>Table</u>	239
<i>AgilePHP :: Table</i>	

<i>Represents a data table in the AgilePHP persistence component.</i>	
Table::addColumn()	240
<i>Pushes a new column to the table's column stack.</i>	
Table::getDisplayNameByProperty()	241
<i>Returns the 'display' attribute value as configured in persistence.xml for the specified 'property' value.</i>	
Table::getForeignKeyColumns()	241
<i>Returns a list of foreign key columns in the current table.</i>	
Table::getPrimaryColumns()	243
<i>Returns a list of primary key columns in the current table.</i>	
Table::getSelectableColumns()	243
Table::getName()	242
<i>Returns the name of the database table</i>	
Table::getModelInstance()	242
<i>Returns an instance of the model as configured in persistence.xml for this 'Table'.</i>	
Table::getForeignKeyColumnsByKey()	242
<i>Returns a list of all foreign keys which share the same 'name' attribute value</i>	
Table::getModel()	242
<i>Returns the name of the domain model responsible for the table</i>	
Table.php	201
<i>AgilePHP Framework :: The Rapid "for developers" PHP5 framework</i>	
<i>Copyright (C) 2009 Make A Byte, inc</i>	

U

User::setCreated()	136
User::setEmail()	136
User::getUsername()	136
User::getSessionId()	136
User::getRoles()	135
User::getSession()	135
User::setLastLogin()	136
User::setPassword()	137
User::setSessionId()	138
User::setUsername()	138
User::setSession()	138
User::setRoles()	138
User::setRole()	137
User::setRoleId()	137
User::getRoleId()	135
User::getRole()	135
Upload::save()	88
<i>Saves the upload contained in the \$_FILES array for the specified file input \$name.</i>	
Upload::setDirectory()	89
<i>Sets the destination directory for the upload.</i>	
Upload::getName()	88
<i>Returns the file form name.</i>	
Upload::getDirectory()	88
<i>Returns the destination directory for uploads.</i>	
Upload	87
<i>AgilePHP :: Upload</i>	

File upload component	
Upload::delete()	88
Deletes the uploaded file.	
Upload::setName()	89
Sets the name used in the file form field element.	
User.php	120
AgilePHP Framework :: The Rapid "for developers" PHP5 framework	
Copyright (C) 2009 Make A Byte, inc	
User::getLastLogin()	135
User::getPassword()	135
User::getEmail()	134
User::getCreated()	134
User	134
AgilePHP :: User	
User domain model	
Upload.php	15
AgilePHP Framework :: The Rapid "for developers" PHP5 framework	
Copyright (C) 2009 Make A Byte, inc	

V

Validator::validateNumber()	91
Validates a number by ensuring it is either an int or float.	
Validator::validateString()	91
Validates the specified data by ensuring it is a string and its length is less than or equal to that of the specified \$length.	
Validator::validateIP()	90
Validates the specified data by ensuring it is a valid IP address.	
Validator::validateEmail()	90
Validates an email address by checking its syntax and performing and DNS lookup on the domain.	
Validator	89
AgilePHP :: Validator	
Provides basic data validation	
Validator::getInstance()	90
Factory method which returns a singleton instance of 'Validator'.	
Validator.php	16
AgilePHP Framework :: The Rapid "for developers" PHP5 framework	
Copyright (C) 2009 Make A Byte, inc	

X

XSLTRenderer::renderXslFile()	192
Renders a view by performing an XSLT transformation. The XSL document is read in from the specified xsl view. The XML document is passed in as a string.	
XSLTRenderer::transform()	192
Performs an XSLT transformation and returns the rendered HTML.	
XSLTRenderer::render()	191
Renders a view by performing an XSLT transformation. The XSL document is passed in as a string. The XML document is passed in as a string.	
XSLTRenderer::getRenderedXslFile()	191

Performs an XSLT tranformation using a specified XSL view document and returns the HTML result.

[XSLTRenderer](#) 190

AgilePHP :: MVC XSLTRenderer

Provides base implementation for XSLT transformations

[XSLTRenderer::ErrorHandler\(\)](#) 191

loadXml reports an error instead of throwing an exception when the xml is not well formed. This

is a custom PHP error handling function which throws an AgilePHP_Exception instead of reporting a PHP error.

[XSLTRenderer.php](#) 165

AgilePHP Framework :: The Rapid "for developers" PHP5 framework

Copyright (C) 2009 Make A Byte, inc

[__autoload_interceptions\(\)](#) 3

[__autoload_class\(\)](#) 3

Lazy loads standard framework and web application classes that do not contain interceptor annotations.

[__autoload\(\)](#) 3

Responsible for 'lazy loading' classes. If agilephp.xml contains a valid <interceptors/> configuration, the loaded class is handed off to __autoload_interceptions.