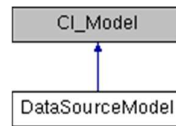


## DataSourceModel Class Reference

Inheritance diagram for DataSourceModel:



### Public Member Functions

**\_\_construct ()**  
**addDataSourceApp** (\$userID, \$dataSource=NULL)  
**addAdvisor** (\$sourceID, \$advisorUsername)  
**askAccess** (\$sourceID, \$userID)  
**addProject** (\$sourceID, \$projectName)  
**askAccessProject** (\$sourceID, \$projectID)  
**deleteDataSource** (\$dataSourceID)  
**deleteAllDataSourcesUser** (\$userID)  
**revokeAccess** (\$sourceID, \$advisorID)  
**updateDataSource** (\$dataSourceID, \$userID=NULL, \$dataSource=NULL)  
**acceptAccess** (\$sourceID, \$advisorID)  
**refuseAccess** (\$sourceID, \$advisorID)  
**acceptAccessProject** (\$sourceID, \$projectID)  
**refuseAccessProject** (\$sourceID, \$projectID)  
**getVisibility** (\$dataSourceID)  
**getDataSource** (\$dataSourceID)  
**searchDataSources** (\$filter=NULL, \$and=false)  
**getPersonalDataSources** (\$userID)  
**getProjects** (\$sourceID)  
**getAccessDataSources** (\$advisorID)  
**getAdvisors** (\$sourceID)  
**getAccessibleDataSources** (\$userID)  
**getUserID** ()  
**getDataSourceID** ()  
**getProjectID** ()  
**getAdvisorID** ()

---

### Constructor & Destructor Documentation

**\_\_construct ()**

**\_\_construct()** is the constructor which initiates the connection to the database

---

### Member Function Documentation

**acceptAccess ( \$sourceID, \$advisorID)**

**acceptAccess()** is a method for accepting the access of a user to a data source

**Parameters:**

<i>\$sourceID</i>	contains the id of the the id of the data source
<i>\$advisorID</i>	is the ID of the user

**acceptAccessProject ( \$sourceID, \$projectID)**

**acceptAccessProject()** is a method for accepting the access of a project to a data source

**Parameters:**

<i>\$sourceID</i>	contains the id of the the id of the data source
<i>\$projectID</i>	is the ID of the project

**addAdvisor ( *\$sourceID*, *\$advisorUsername* )**

**addAdvisor()** is a method for adding a data source for a user

**Parameters:**

<i>\$sourceID</i>	(required) is the id of a data source
<i>\$advisorUsername</i>	(required) is the login of a user

**Returns:**

the affected rows

**addDataSourceApp ( *\$userID*, *\$dataSource* = NULL )**

**addDataSourceApp()** is a method for adding a file

**Parameters:**

<i>\$userID</i>	is is the id of the owner of the data source
<i>\$dataSource</i>	is an array containing the informations about the data source
<i>\$dataSource</i> ['name']	(required) is the name of the data source
<i>\$dataSource</i> ['url']	(optional) is the access url to the data source (default value=NULL)
<i>\$dataSource</i> ['appli']	(optional) means if the id is an application (1) or not (0) =default
<i>\$dataSource</i> ['config']	(optional) could be, for exemple, a JSON app configuration file
<i>\$dataSource</i> ['visible']	(required) means if the data source have to be hidden (=2), visible (=1) or on demand (=0, default)

**Returns:**

new data source id if insert succeeded and FALSE if not

**addProject ( *\$sourceID*, *\$projectName* )**

**addProject()** is a method for adding a data source for a user

**Parameters:**

<i>\$sourceID</i>	(required) is the id of a data source
<i>\$projectName</i>	(required) is the login of a user

**Returns:**

the affected rows

**askAccess ( *\$sourceID*, *\$userID* )**

**askAccess()** is a method for asking the access to a data source

**Parameters:**

<i>\$sourceID</i>	(required) is the id of a data source
<i>\$userID</i>	(required) is the id of a user

**askAccessProject ( *\$sourceID*, *\$projectID* )**

**askAccessProject()** is a method for asking the access to a data source

**Parameters:**

<i>\$sourceID</i>	(required) is the id of a data source
<i>\$projectID</i>	(required) is the id of a user

**deleteAllDataSourcesUser ( *\$userID* )**

**deleteAllDataSourcesUser()** remove all data sources for a specific user

**Parameters:**

<i>\$userID</i>	
-----------------	--

**Returns:**

a boolean (TRUE if deletion has been applied, FALSE if not)

**deleteDataSource ( \$dataSourceID)**

**deleteDataSource()** delete a data source based on its id

**Parameters:**

<i>\$dataSourceID</i>	
-----------------------	--

**Returns:**

a boolean (TRUE if deletion has been applied, FALSE if not)

**getAccessDataSources ( \$advisorID)**

**getAccessDataSources()** is a method to search for data sources that a user has access to

**Parameters:**

<i>\$advisorID</i>	is the user ID
--------------------	----------------

**Returns:**

an array of data sources

\$response['id'] is the data source id

\$response['file\_name'] is the name of the data source

\$response['url'] is the URL of the data source

\$response['application'] 0 for file and 1 for application

\$response['configuration'] is the configuration File

\$response['visible'] default visibility attribute for user files (0=hidden, 1=visible, 2=on demand)

\$response['add\_date'] is the creation date of the data source in the database

**getAccessibleDataSources ( \$userID)**

**getAccessibleDataSources()** is a method to search for users who requested access to data sources

**Parameters:**

<i>\$userID</i>	is the user ID
-----------------	----------------

**Returns:**

an array of data sources

\$response['id'] is the data source id

\$response['owner\_id'] is the owner id

\$response['name'] is the name of the data source

\$response['is\_application'] describes whether the data source is an application or not

\$response['visibility'] is the visibility of the data source (0, 1, 2)

\$response['published'] is the date when the data source was added

**getAdvisorID ()**

**getAdvisorID()** is a method for searching the advisorID of the last record

**Returns:**

advisorID

**See also:**

function for tests

**getAdvisors ( \$sourceID)**

**getAdvisors()** is a method to search for users who requested access to data sources

**Parameters:**

<i>\$sourceID</i>	is the data source ID
-------------------	-----------------------

**Returns:**

an array of users

\$response['userid'] is the user id

\$response['username'] is the user login

\$response['firstname'] is the user firstname

\$response['lastname'] is the user lastname  
\$response['state'] is the state of accessibility to the data source

### **getDataSource ( \$dataSourceID)**

**getDataSource()** this method returns a data source based on its id

#### **Parameters:**

<i>\$dataSourceID</i>	data source id
-----------------------	----------------

#### **Returns:**

a data source with its informations  
\$response['id'] is the data source id  
\$response['owner\_id'] is the owner id  
\$response['name'] is the name of the data source  
\$response['url'] is the url of the data source  
\$response['application'] 0 for file and 1 for application  
\$response['configuration'] is the configuration File  
\$response['visible'] default visibility attribute for user files (0=hidden, 1=visible, 2=on demand)  
\$response['add\_date'] is the creation date of the data source in the database

### **getDataSourceID ()**

**getDataSourceID()** is a method for searching the dataSourceID of the last record

#### **Returns:**

dataSourceID

#### **See also:**

function for tests

### **getPersonalDataSources ( \$userID)**

**getPersonalDataSources()** this method returns the data sources that belong to a user

#### **Parameters:**

<i>\$userID</i>	user id
-----------------	---------

#### **Returns:**

datasources of the user  
\$response['id'] is the data source id  
\$response['file\_name'] is the name of the data source  
\$response['url'] is the URL of the data source  
\$response['application'] 0 for file and 1 for application  
\$response['configuration'] is the configuration File  
\$response['visible'] default visibility attribute for user files (0=hidden, 1=visible, 2=on demand)  
\$response['add\_date'] is the creation date of the data source in the database

### **getProjects ( \$sourceID)**

**getProjects()** is a method for searching the projects linked with a data source in the database

#### **Parameters:**

<i>\$sourceID</i>	data source id
-------------------	----------------

#### **Returns:**

an array of projects that requested access to a data source  
\$response['id'] is the project id  
\$response['name'] is the name of the project  
\$response['end\_date'] is the date of the end of the project  
\$response['state'] is the 'access state' with the possible values 0(=asked), 1(=accepted), 2(=refused)

### **getProjetID ()**

**getProjetID()** is a method for searching the ProjetID of the last record

**Returns:**

projctID

**See also:**

function for tests

**getUserID ()**

**getUserID()** is a method for searching the userID of the last record

**Returns:**

userID

**See also:**

function for tests

**getVisibility ( \$dataSourceID)**

**getVisibility()** this method returns a data source based on its id

**Parameters:**

<i>\$dataSourceID</i>	data source id
-----------------------	----------------

**Returns:**

the value of the visibility of a data source with its informations

**refuseAccess ( \$sourceID, \$advisorID)**

**refuseAccess()** is a method for refusing the access of a user to a data source

**Parameters:**

<i>\$sourceID</i>	contains the id of the the id of the data source
<i>\$advisorID</i>	is the ID of the user

**refuseAccessProject ( \$sourceID, \$projectID)**

**refuseAccessProject()** is a method for refusing the access of a project to a data source

**Parameters:**

<i>\$projectID</i>	is the id of the user invited to the data source
<i>\$sourceID</i>	is the data source id

**revokeAccess ( \$sourceID, \$advisorID)**

**revokeAccess()** remove a data source for a specific user

**Parameters:**

<i>\$adivsorID</i>	is the id of the user invited to the data source
<i>\$sourceID</i>	is the data source id

**searchDataSources ( \$filter = NULL, \$and = false)**

**searchDataSources()** is a method for searching data sources in the database

**Parameters:**

<i>\$filter</i>	is optional and is an array containing search criterions
<i>\$filter['owner']</i>	is optional and contains the name (can be partial) of the owner of a data source
<i>\$filter['name']</i>	is optional and contains the name (can be partial) of searched data source(s)

**Returns:**

an array of data sources (ordered by file name)

**See also:**

**searchDataSources()** for the data structure of returned data sources

**updateDataSource ( \$dataSourceID, \$userID = NULL, \$dataSource = NULL)**

**updateDataSource()** is a method for updating a specific data source

**Parameters:**

<i>\$userID</i>	is is the id of the owner of the data source
<i>\$dataSource</i>	is an array containing the informations about the data source
<i>\$dataSourceID</i>	is the id of the data source
<i>\$dataSource</i> ['name']	(optional) is the new name of the data source
<i>\$dataSource</i> ['url']	(optional) is the new access url to the data source (default value=NULL)
<i>\$dataSource</i> ['appli']	(optional) means if the id is an application (1) or not (0) =default
<i>\$dataSource</i> ['config']	(optional) could be, for exemple, a JSON app configuration file
<i>\$dataSource</i> ['visible']	(optional) means if the data source have to be hidden (=2), visible (=1) or if it is accessible on request (=0)

**Returns:**

TRUE if update succeeded and FALSE if not