

Data Structure Documentation

UserModel Class Reference (v 1.0)

Inherits CI_Model.

Public Member Functions

- **__construct** ()
- **authentication** (\$login, \$password)
- **get_user_roles** (\$userid)
- **get_roles** ()
- **get_role** (\$roleid)
- **get_user_rights** (\$userid)
- **get_rights** ()
- **get_role_rights** (\$roleid)
- **get_right** (\$rightid)
- **get_user** (\$userid)
- **get_users** (\$filter=NULL, \$and=false)
- **login_is_free** (\$login)
- **get_users_from_role** (\$roleid)
- **get_advice** (\$advid)
- **get_advices** (\$filter=NULL)
- **get_user_advices** (\$userid)
- **get_advisor_advices** (\$userid)
- **add_user** (\$user=NULL)
- **add_user_role** (\$userid, \$roleid)
- **add_role** (\$name, \$description=NULL)
- **add_right** (\$name, \$description=NULL)
- **add_role_right** (\$roleid, \$rightid)
- **add_advice** (\$user_id, \$advisor_id, \$advice)
- **update_user** (\$user=NULL)
- **update_role** (\$id, \$name=NULL, \$description=NULL)
- **update_right** (\$id, \$name=NULL, \$description=NULL)
- **update_advice** (\$id, \$user_id=NULL, \$advisor_id=NULL, \$advice=NULL)
- **delete_user** (\$userid)
- **delete_user_role** (\$userid, \$roleid)
- **delete_user_all_role** (\$userid)
- **delete_role** (\$roleid)
- **delete_right** (\$rightid)
- **delete_role_right** (\$roleid, \$rightid)
- **delete_role_all_right** (\$roleid)
- **delete_advice** (\$advid)

Constructor & Destructor Documentation

__construct ()

__construct() is the constructor which initiates the connection to the database

Member Function Documentation

add_advice (\$user_id, \$advisor_id, \$advice)

add_advice() is a method for adding an advice

Parameters:

<i>\$userid</i>	(required) id of advised user
<i>\$advisor_id</i>	(required) id of advisor
<i>\$advice</i>	(required) advice text

Returns:

new advice id if insert succeeded and FALSE if not

add_right (\$name, \$description = NULL)

add_right() is a method for adding a right

Parameters:

<i>\$name</i>	(required) right name (must be unique)
<i>\$description</i>	(optional) right description

Returns:

new right id if insert succeeded and FALSE if not

add_role (\$name, \$description = NULL)

add_role() is a method for adding a role

Parameters:

<i>\$name</i>	(required) role name (must be unique)
<i>\$description</i>	(optional) role description

Returns:

new role id if insert succeeded and FALSE if not

add_role_right (\$roleid, \$rightid)

add_role_right() is a method for adding a right to a role

Parameters:

<i>\$roleid</i>	
<i>\$rightid</i>	

Returns:

TRUE if insert succeeded and FALSE if not

add_user (\$user = NULL)

add_user() is a method for adding an user

Parameters:

<i>\$user</i>	is an array containing user data
<i>\$user['lastname']</i>	(required) contains the lastname
<i>\$user['firstname']</i>	(required) contains the firstname
<i>\$user['birthdate']</i>	(optional) contains the birthdate
<i>\$user['email']</i>	(optional) contains the email address
<i>\$user['phone']</i>	(optional) contains the phone number
<i>\$user['mobile']</i>	(optional) contains the mobile phone number
<i>\$user['gender']</i>	(optional) 0 for female and 1 for male
<i>\$user['login']</i>	(required) contains the login
<i>\$user['password']</i>	(required) contains the password, it will be hashed inside this method with a SHA1 function
<i>\$user['visible']</i>	(optional) contains the default visibility attribute for user files

	(0=hidden=default, 1=visible, 2=on demand)
<i>\$user['advice']</i>	(optional) means if a user is ok to receive advices (1) or not (0) =default

Returns:

new user id if insert succeeded and FALSE if not

add_user_role (*\$userid*, *\$roleid*)

add_user_role() is a method for adding a role to an user

Parameters:

<i>\$userid</i>	
<i>\$roleid</i>	

Returns:

TRUE if insert succeeded and FALSE if not

authentication (*\$login*, *\$password*)

authentication() is a method for querying if an user have or not an account

Parameters:

<i>\$login</i>	
<i>\$password</i>	(will be hashed inside this method with a SHA1 function)

Returns:

an array containing user informations

See also:

get_user() for the data structure of returned user

delete_advice (*\$advid*)

delete_advice() delete an advice based on its id

Parameters:

<i>\$advid</i>	advice id
----------------	-----------

Returns:

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

delete_right (*\$rightid*)

delete_right() delete a right based on its id

Parameters:

<i>\$rightid</i>	
------------------	--

Returns:

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

delete_role (*\$roleid*)

delete_role() delete a role based on its id

Parameters:

<i>\$roleid</i>	
-----------------	--

Returns:

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

delete_role_all_right (*\$roleid*)

delete_role_all_right() remove all rights for a specific role

Parameters:

<i>\$roleid</i>	
-----------------	--

Returns:

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

delete_role_right (\$roleid, \$rightid)

delete_role_right() remove a right for a specific role

Parameters:

<i>\$roleid</i>	
<i>\$rightid</i>	

Returns:

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

delete_user (\$userid)

delete_user() delete an user based on its id

Parameters:

<i>\$userid</i>	
-----------------	--

Returns:

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

delete_user_all_role (\$userid)

delete_user_all_role() remove all roles for a specific user

Parameters:

<i>\$userid</i>	
-----------------	--

Returns:

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

delete_user_role (\$userid, \$roleid)

delete_user_role() remove a role for a specific user

Parameters:

<i>\$userid</i>	
<i>\$roleid</i>	

Returns:

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

get_advice (\$advid)

get_advices() this method returns an advice based on its id

Parameters:

<i>\$dvid</i>	advice id
---------------	-----------

Returns:

an advices + firstnames and lasnames of users and advisers
\$response['id'] is the advice id
\$response['user_id'] is the advised user id
\$response['advisor_id'] is the advisor id
\$response['advice'] is the advice text
\$response['advice_date'] is the creation date of the advice
\$response['user_lastname'] is the lastname of advised user
\$response['user_firstname'] is the firstname of advised user
\$response['advisor_lastname'] is the lastname of advisor
\$response['advisor_firstname'] is the firstname of advisor

get_advices (\$filter = NULL)

get_advices() is a method for searching advices in the database

Parameters:

<i>\$filter</i>	is optional and is an array containing search criterions
<i>\$filter['advice']</i>	is optional and contains a substring which will be searched into the advice's text
<i>\$filter['user_id']</i>	is optional and contains the identifier of the advised user
<i>\$filter['advisor_id']</i>	is optional and contains the identifier of the advisor (which is also an user)
<i>\$filter['advice_date']</i>	is optional and contains the creation date (without time) of one or more advices
<i>\$filter['advice_date_start']</i>	is optional and contains the minimal creation date (without time) of one or more advices
<i>\$filter['advice_date_end']</i>	is optional and contains the maximal creation date (without time) of one or more advices

Returns:

an array of advices with firstnames and lasnames of users and advisers. Results are sorted by advice_date (descending)

See also:

get_advices() for the data structure of an advice

get_advisor_advices (\$userid)

get_advisor_advices() is a method for searching all advices written by a specific user (advisor)

Parameters:

<i>\$userid</i>	contains the identifier of the advisor
-----------------	--

Returns:

an array of advices with firstnames and lasnames of users and advisers. Results are sorted by advice_date (descending)

See also:

get_advices() for returned data structure

get_right (\$rightid)

get_right() return a right based on its id

Parameters:

<i>\$rightid</i>	
------------------	--

Returns:

an array containing right informations
\$response['id'] right id
\$response['name'] right name (unique)
\$response['description'] right description

get_rights ()

get_rights() returns all rights

Returns:

an array of rights (name order)

See also:

get_right() for returned data structure

get_role (\$roleid)

get_role() returns role informations based on its id

Parameters:

<i>\$roleid</i>	
-----------------	--

Returns:

an array containing role informations
\$response['id'] role id
\$response['name'] role name (unique)
\$response['description'] role description

get_role_rights (*\$roleid*)

get_rights() returns all the rights corresponding to a specific role

Parameters:

<i>\$roleid</i>	
-----------------	--

Returns:

an array of rights (name order)

See also:

get_right() for returned data structure

get_roles ()

get_roles() returns all roles

Returns:

an array of roles (name order)

See also:

get_role() for returned data structure

get_user (*\$userid*)

get_user() this method returns user informations based on its user id

Parameters:

<i>\$userid</i>	
-----------------	--

Returns:

an array containing user informations
\$response['id'] is the user id
\$response['lastname']
\$response['firstname']
\$response['birthdate']
\$response['email']
\$response['phone']
\$response['mobile']
\$response['gender'] 0 for female and 1 for male
\$response['login']
\$response['visible'] default visibility attribute for user files (0=hidden, 1=visible, 2=on demand)
\$response['advice'] 1 if a user is ok to receive advices and 0 if not

get_user_advices (*\$userid*)

get_user_advices() is a method for searching all advices sended to a specific user

Parameters:

<i>\$userid</i>	contains the identifier of the advised user
-----------------	---

Returns:

an array of advices whith firstnames and lasnames of users and advisers. Results are sorted by advice_date (descending)

See also:

`get_advices()` for returned data structure

get_user_rights (*\$userid*)

`get_user_rights()` returns all the rights of a specific user

Parameters:

<i>\$userid</i>	
-----------------	--

Returns:

an array of rights (name order)

See also:

`get_right()` for returned data structure

get_user_roles (*\$userid*)

`get_user_roles()` returns all roles for a specific user

Parameters:

<i>\$userid</i>	
-----------------	--

Returns:

an array of roles (name order)

See also:

`get_role()` for returned data structure

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

`get_users()` is a method for searching users in the database

Parameters:

<i>\$filter</i>	is optional and is an array containing search criterions
<i>\$filter['lastname']</i>	is optional and contains the lastname (can be partial) of seached user(s)
<i>\$filter['firstname']</i>	is optional and contains the firstname (can be partial) of seached user(s)
<i>\$filter['birthdate']</i>	is optional and contains the birthdate of seached user(s)
<i>\$filter['email']</i>	is optional and contains the email address (can be partial) of seached user(s)
<i>\$filter['phone']</i>	is optional and contains the phone number (can be partial) of seached user(s)
<i>\$filter['mobile']</i>	is optional and contains the mobile phone number (can be partial) of seached user(s)
<i>\$filter['gender']</i>	is optional and is a boolean which is for the gender of searched users (0 = female, 1 is male)
<i>\$filter['login']</i>	is optional and contains the login (can be partial) of seached user(s)
<i>\$filter['visible']</i>	is optional and contains the default visibility attribute for user files (0=hidden, 1=visible, 2=on demand)
<i>\$filter['advice']</i>	is optional and is a boolean which means if a user is ok to receive advices (1) or not (0)
<i>\$and</i>	is optional and is an boolean which is FALSE (default behavior) for processing teh search query with OR operators and TRUE for AND operators

Returns:

an array of users (lastname, firstname ascending order)

See also:

`get_user()` for the data structure of returned users

get_users_from_role (*\$roleid*)

`get_users_from_role()` this method returns all users having a specific role

Parameters:

<i>\$roleid</i>	
-----------------	--

Returns:

an array of users (lastname, firstname ascending order)

See also:

`get_user()` for the data structure of returned users

login_is_free (*\$login*)

`login_is_free()` this method checks if a login is not already used

Parameters:

<i>\$login</i>	
----------------	--

Returns:

a boolean : FALSE if the login is already used, TRUE if not

update_advice (*\$id*, *\$user_id* = NULL, *\$advisor_id* = NULL, *\$advice* = NULL)

`update_advice()` is a method for updating a specific advice

Parameters:

<i>\$id</i>	advice id (required)
<i>\$user_id</i>	new user id of advised user (optional)
<i>\$advisor_id</i>	new user id of advisor (optional)
<i>\$advice</i>	new advice text (optional)

Returns:

TRUE if update succeeded and FALSE if not

update_right (*\$id*, *\$name* = NULL, *\$description* = NULL)

`update_right()` is a method for updating name and/or description of a specific right

Parameters:

<i>\$id</i>	right id (required)
<i>\$name</i>	new right name (optional)
<i>\$description</i>	new right description (optional)

Returns:

TRUE if update succeeded and FALSE if not

update_role (*\$id*, *\$name* = NULL, *\$description* = NULL)

`update_role()` is a method for updating name and/or description of a specific role

Parameters:

<i>\$id</i>	role id (required)
<i>\$name</i>	new role name (optional)
<i>\$description</i>	new role description (optional)

Returns:

TRUE if update succeeded and FALSE if not

update_user (*\$user* = NULL)

`update_user()` is a method for updating informations of a specific user

Parameters:

<i>\$user['id']</i>	(required) user id of the user whose informations need to be updated
<i>\$user['lastname']</i>	is optional and contains the lastname
<i>\$user['firstname']</i>	is optional and contains the firstname
<i>\$user['birthdate']</i>	is optional and contains the birthdate
<i>\$user['email']</i>	is optional and contains the email address
<i>\$user['phone']</i>	is optional and contains the phone number
<i>\$user['mobile']</i>	is optional and contains the mobile phone number

<i>\$user['gender']</i>	is optional, 0 for female and 1 for male
<i>\$user['login']</i>	is optional and contains the login
<i>\$user['visible']</i>	is optional and contains the default visibility attribute for user files (0=hidden, 1=visible, 2=on demand)
<i>\$user['advice']</i>	is optional and means if a user is ok to receive advices (1) or not (0)

Returns:

TRUE if update succeeded and FALSE if not

The documentation for this class was generated from the following file:

- UserModel.php

Index

INDEX