

WaterScrum

V0.0.3

Generated by Doxygen 1.8.13

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/backlog.php File Reference	3
2.1.1	Detailed Description	3
2.1.2	Function Documentation	4
2.1.2.1	add_inside_project_role()	4
2.1.2.2	add_inside_project_US()	4
2.1.2.3	get_all_inside_project_role()	5
2.1.2.4	get_all_US_by_project_id()	5
2.1.2.5	modify_inside_project_role()	5
2.1.2.6	modify_inside_project_US()	6
2.1.2.7	remove_by_role_id()	7
2.1.2.8	remove_US_by_id()	7
2.2	/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/doc.php File Reference . .	7
2.2.1	Detailed Description	8
2.2.2	Function Documentation	8
2.2.2.1	add_new_doc()	8
2.2.2.2	all_doc_done()	9
2.2.2.3	change_state_doc()	9
2.2.2.4	check_deprecated_doc()	10
2.2.2.5	compute_proportion_doc()	10
2.2.2.6	count_proportion_doc()	10

2.2.2.7	delete_doc()	11
2.2.2.8	edit_doc()	11
2.2.2.9	get_all_deprecated_doc()	12
2.2.2.10	get_all_done_doc()	12
2.2.2.11	get_all_todo_doc()	13
2.3	/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/faq.php File Reference	13
2.3.1	Detailed Description	14
2.3.2	Function Documentation	14
2.3.2.1	delQA()	14
2.3.2.2	editQA()	14
2.3.2.3	getCategories()	14
2.3.2.4	getCategoryId()	15
2.3.2.5	getQA()	15
2.3.2.6	searchQA()	15
2.4	/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/logout.php File Reference	16
2.4.1	Detailed Description	16
2.5	/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/modPanel.php File Reference	16
2.5.1	Detailed Description	17
2.5.2	Function Documentation	17
2.5.2.1	changeUserRole()	17
2.5.2.2	deleteUser()	17
2.5.2.3	getAllUsers()	17
2.6	/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/profile.php File Reference	19
2.6.1	Detailed Description	19
2.6.2	Function Documentation	19
2.6.2.1	AcceptInvitation()	19
2.6.2.2	CancelRequest()	20
2.6.2.3	deleteAccount()	20
2.6.2.4	editInfo()	21
2.6.2.5	getUserInvitationsOrRequest()	21

2.6.2.6	getUserNbParticipation()	21
2.6.2.7	getUserProfile()	22
2.7	/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/projects.php File Reference	22
2.7.1	Detailed Description	23
2.7.2	Function Documentation	23
2.7.2.1	add_invitation_request()	23
2.7.2.2	associat_project_and_user()	23
2.7.2.3	check_if_request_already_exist()	24
2.7.2.4	create_new_project()	24
2.7.2.5	get_all_project_by_user_id()	25
2.7.2.6	get_all_project_without_user_id()	25
2.7.2.7	leave_a_project()	26
2.7.2.8	remove_by_project_id()	26
2.8	/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/release.php File Reference	27
2.8.1	Detailed Description	27
2.8.2	Function Documentation	27
2.8.2.1	change_git_url()	27
2.8.2.2	delete_all_commits()	28
2.8.2.3	deprecate_all_test()	28
2.8.2.4	get_all_commit()	29
2.8.2.5	get_git_url()	29
2.8.2.6	get_last_commit()	29
2.8.2.7	save_all_commit()	30
2.9	/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/selectedProject.php File Reference	30
2.9.1	Detailed Description	31
2.9.2	Function Documentation	31
2.9.2.1	acceptRequest()	31
2.9.2.2	deleteInvitationOrRequest()	31
2.9.2.3	deleteMember()	33
2.9.2.4	get_all_project_invitations()	33

2.9.2.5	get_all_project_joining_requests()	34
2.9.2.6	get_all_project_members()	34
2.9.2.7	get_project_by_id()	34
2.9.2.8	get_project_master()	35
2.9.2.9	is_master()	35
2.9.2.10	is_member()	36
2.10	/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/sprints.php File Reference	36
2.10.1	Detailed Description	37
2.10.2	Function Documentation	37
2.10.2.1	add_sprint_background()	37
2.10.2.2	count_nb_task_by_state_in_sprint()	38
2.10.2.3	create_new_sprint()	38
2.10.2.4	create_new_task()	39
2.10.2.5	delete_sprint_by_id()	39
2.10.2.6	get_all_project_members_and_master()	40
2.10.2.7	get_all_sprints()	40
2.10.2.8	get_all_task_inside_sprint()	40
2.10.2.9	get_all_us_inside_project()	41
2.10.2.10	get_all_us_inside_sprint()	41
2.10.2.11	get_all_us_inside_task()	42
2.10.2.12	link_us_to_sprint()	42
2.10.2.13	link_us_to_task()	42
2.10.2.14	remove_task()	43
2.10.2.15	switch_task_state()	43
2.10.2.16	unlink_us_from_sprint()	44
2.10.2.17	unlink_us_from_task()	44
2.10.2.18	update_task()	45
2.11	/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/tests.php File Reference	45
2.11.1	Detailed Description	46
2.11.2	Function Documentation	46

2.11.2.1	add_new_test()	46
2.11.2.2	change_state()	47
2.11.2.3	check_deprecated()	47
2.11.2.4	compute_proportion()	47
2.11.2.5	count_proportion()	48
2.11.2.6	delete_test()	48
2.11.2.7	edit_test()	49
2.11.2.8	get_all_deprecated_tests()	49
2.11.2.9	get_all_failed_tests()	50
2.11.2.10	get_all_never_run_tests()	50
2.11.2.11	get_all_passed_tests()	50
2.11.2.12	pass_all_tests()	51
2.12	/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/utility.php File Reference	51
2.12.1	Detailed Description	52
2.12.2	Function Documentation	52
2.12.2.1	add_invitation_by_user_name()	52
2.12.2.2	check_if_request_already_exist()	52
2.12.2.3	get_all_user_not_in_project()	53
	Index	55

Chapter 1

File Index

1.1 File List

Here is a list of all documented files with brief descriptions:

/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ backlog.php	
Various functions, who have for objectif to give all the necessary for the controller backlog you will find fonctions for userStories insideProjectRole essentially CRUD function	3
/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ doc.php	
Various functions to make the doc page works Mainly CRUD functions, plus some to deal with the progression bar	7
/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ faq.php	
Various functions to communicate with the faq and faq_comments table of the database. Essentially CRUD functions	13
/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ index.php	??
/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ login.php	??
/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ logout.php	
This file contains only one function that allows you to disconnect from the website	16
/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ modPanel.php	
Various functions to enable administrator monitoring from the mod panel page. Mainly CRUD functions	16
/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ profile.php	
Various functions, who have for objectif to give all the necessary for the controller profile you will find fonctions for project_invitation user essentially CRUD function	19
/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ projects.php	
Various functions, who have for objectif to give all the necessary for the controller projects you will find fonctions for project user inside project essentially CRUD function	22
/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ release.php	
Various functions, who have for objectif to give all the necessary for the controller release you will find fonctions for project_commits essentially CRUD function	27
/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ selectedProject.php	
Various functions, who have for objectif to give all the necessary for selectedProject you will find fonctions for project user essentially CRUD function	30
/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ signup.php	??
/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ sprints.php	
Various functions, who have for objectif to give all the necessary for sprint controller you will find fonctions for project user essentially CRUD function	36
/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ tests.php	
Various functions to make the tests page works Mainly CRUD functions, plus some to deal with the progression bar	45
/autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/ utility.php	
Various functions, who have for objectif to give all the necessary for all controller you will find fonctions for project user essentially CRUD function	51

Chapter 2

File Documentation

2.1 /autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/backlog.php File Reference

Various functions, who have for objectif to give all the necessary for the controller backlog you will find fonctions for userStories insideProjectRole essentially CRUD function.

Functions

- [add_inside_project_role](#) (\$projectId, \$roleName, \$roleDescription)
this function add a given project_role inside database on inside_project_role table
- [get_all_inside_project_role](#) (\$projectId)
this function return all inside_project_role of a given project
- [modify_inside_project_role](#) (\$roleId, \$roleName, \$roleDescription)
this function modify a given project_role inside database on inside_project_role table
- [remove_by_role_id](#) (\$id)
this function remove a given project_role inside database on inside_project_role table
- [get_all_US_by_project_id](#) (\$project_id)
this function return all User stories of a given project
- [add_inside_project_US](#) (\$projectId, \$USName, \$roleId, \$iCan, \$soThat, \$difficulty, \$workValue, \$done)
this function add a given user_story inside database on user_story table
- [remove_US_by_id](#) (\$USId)
this function remove a given user_story inside database on user_story table
- [modify_inside_project_US](#) (\$usID, \$projectId, \$USName, \$roleId, \$iCan, \$soThat, \$difficulty, \$workValue, \$done)
this function modify a given user_story inside database on user_story table

2.1.1 Detailed Description

Various functions, who have for objectif to give all the necessary for the controller backlog you will find fonctions for userStories insideProjectRole essentially CRUD function.

Backlog

2.1.2 Function Documentation

2.1.2.1 add_inside_project_role()

```
add_inside_project_role (
    $projectID,
    $roleName,
    $roleDescription )
```

this function add a given project_role inside database on inside_project_role table

Parameters

<i>projectID</i>	the project id where you want to insert the project_role
<i>roleName</i>	the name of the role that you want add to the project
<i>roleDescription</i>	the description of the role that you want add to the project

Returns

The last insert index on the database or -1 if an exception occurs

Definition at line 19 of file backlog.php.

2.1.2.2 add_inside_project_US()

```
add_inside_project_US (
    $projectID,
    $USName,
    $roleId,
    $iCan,
    $soThat,
    $difficulty,
    $workValue,
    $done )
```

this function add a given user_story inside database on user_story table

Parameters

<i>projectID</i>	the project id where you want to insert the user_story
<i>USName</i>	the name of the User_story that you want add to the project
<i>roleId</i>	the role id that you want to link with the user_story
<i>iCan</i>	the iCan description of the user_story that you want add to the project
<i>soThat</i>	the soThat description of the user_story that you want add to the project
<i>difficulty</i>	the difficulty of the US value possible : 1 2 3 5 8 13 21 34
<i>workValue</i>	the workValue of the US value possible : low medium hight very-high
<i>done</i>	a boolean integer that indicates if the user story is finished

Returns

The last insert index on the database or -1 if an exception occurs

Definition at line 150 of file backlog.php.

2.1.2.3 get_all_inside_project_role()

```
get_all_inside_project_role (
    $projectID )
```

this function return all inside_project_role of a given project

Parameters

<i>projectID</i>	the project id where you want to get the project_role
------------------	---

Returns

the PDOStatement contening all the project role or return -1 if an exception occurs

Definition at line 45 of file backlog.php.

2.1.2.4 get_all_US_by_project_id()

```
get_all_US_by_project_id (
    $project_id )
```

this function return all User stories of a given project

Parameters

<i>projectID</i>	the project id where you want to get the user stories
------------------	---

Returns

the PDOStatement contening all the userStories or return -1 if an exception occurs

Definition at line 120 of file backlog.php.

2.1.2.5 modify_inside_project_role()

```
modify_inside_project_role (
    $roleID,
```

```

    $roleName,
    $roleDescription )

```

this function modify a given project_role inside database on inside_project_role table

Parameters

<i>roleID</i>	the role id that you want to modify
<i>roleName</i>	the new name to insert inside the role
<i>roleDescription</i>	the new description to insert inside the role

Returns

the roleID that you have modify or -1 if an exception occurs

Definition at line 71 of file backlog.php.

2.1.2.6 modify_inside_project_US()

```

modify_inside_project_US (
    $usID,
    $projectID,
    $USName,
    $roleId,
    $iCan,
    $soThat,
    $difficulty,
    $workValue,
    $done )

```

this function modify a given user_story inside database on user_story table

Parameters

<i>usID</i>	the user_story id that you want to modify
<i>USName</i>	the new value for the name
<i>roleId</i>	the new value for the roleID
<i>iCan</i>	the new value for the iCan
<i>soThat</i>	the new value for the soThat
<i>difficulty</i>	the new value for the difficulty value possible : 1 2 3 5 8 13 21 34
<i>workValue</i>	the new value for the workValue value possible : low medium hight very-hight
<i>done</i>	the new value for the done

Returns

The id of the US or -1 if an exception occurs

Definition at line 212 of file backlog.php.

2.1.2.7 remove_by_role_id()

```
remove_by_role_id (
    $id )
```

this function remove a given project_role inside database on inside_project_role table

Parameters

<i>roleID</i>	the role id that you want to remove
---------------	-------------------------------------

Returns

1 if succes -1 if an exception occurs

Definition at line 98 of file backlog.php.

2.1.2.8 remove_US_by_id()

```
remove_US_by_id (
    $USId )
```

this function remove a given user_story inside database on user_story table

Parameters

<i>roleID</i>	the user_story id that you want to remove
---------------	---

Returns

1 if succes -1 if an exception occurs

Definition at line 184 of file backlog.php.

2.2 /autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/doc.php File Reference

Various functions to make the doc page works Mainly CRUD functions, plus some to deal with the progression bar.

Functions

- [get_all_done_doc](#) (\$projectId)
Get all the doc that is marked as 'done' in the database.
- [get_all_todo_doc](#) (\$projectId)

- Get all the doc that is marked as 'todo' in the database.*

 - [get_all_deprecated_doc](#) (\$projectId)
- Get all the doc that is marked as 'deprecated' in the database.*

 - [count_proportion_doc](#) (\$projectId)
- Compute proportion of each type of doc (done, to do, deprecated)*

 - [compute_proportion_doc](#) (\$projectId)
- Compute the doc proportions of each category (done, to do, deprecated) to make percentages.*

 - [add_new_doc](#) (\$projectId, \$name, \$description, \$state)
- Function to add a new doc element to the specified project.*

 - [all_doc_done](#) (\$projectId)
- Function to mark all doc elements of a given project as done and refresh their last update date.*

 - [edit_doc](#) (\$id, \$name, \$description, \$state)
- Modify the informations of a doc from its id.*

 - [delete_doc](#) (\$id, \$state)
- Delete a doc element with its id.*

 - [change_state_doc](#) (\$id, \$state)
- Change the state of a doc element from its id and refresh its last update date. The function can be used only to refresh last update date (without state change)*

 - [check_deprecated_doc](#) (\$projectId)
- This function check all done doc in the database for a given project and mark it as deprecated if it is too old.*

2.2.1 Detailed Description

Various functions to make the doc page works Mainly CRUD functions, plus some to deal with the progression bar.

Doc

2.2.2 Function Documentation

2.2.2.1 add_new_doc()

```
add_new_doc (
    $projectId,
    $name,
    $description,
    $state )
```

Function to add a new doc element to the specified project.

Parameters

<i>projectId</i>	The id of the project we want to add the doc to
<i>name</i>	The name of the doc
<i>description</i>	The text description of the doc
<i>state</i>	The enum state of the doc (can be done, to do or deprecated)

Returns

state Return the state of the added doc(this state will be used by the jQuery code to know which part of the page to refresh)

Definition at line 157 of file doc.php.

2.2.2.2 all_doc_done()

```
all_doc_done (
    $projectId )
```

Function to mark all doc elements of a given project as done and refresh their last update date.

Parameters

<i>projectId</i>	The id of the project that will have all its doc marked as done
------------------	---

Returns

stmt The statement value of the executed sql instruction

Definition at line 186 of file doc.php.

2.2.2.3 change_state_doc()

```
change_state_doc (
    $id,
    $state )
```

Change the state of a doc element from its id and refresh its last update date. The function can be used only to refresh last update date (without state change)

Parameters

<i>id</i>	The id of the doc that will change state
<i>state</i>	The new state of the doc

Returns

stmt The returned statement of the executed sql request

Definition at line 266 of file doc.php.

2.2.2.4 check_deprecated_doc()

```
check_deprecated_doc (
    $projectId )
```

This function check all done doc in the database for a given project and mark it as deprecated if it is too old.

Parameters

<i>project↔ Id</i>	The id of the project we want to check deprecated doc from
------------------------	--

Returns

number Return the number of deprecated doc element found. If there was an error during execution, return -1

Definition at line 293 of file doc.php.

2.2.2.5 compute_proportion_doc()

```
compute_proportion_doc (
    $projectId )
```

Compute the doc proportions of each category (done, to do, deprecated) to make percentages.

Parameters

<i>project↔ Id</i>	The id of the project we want to compute the doc proportions from
------------------------	---

Returns

percentages Return an array containing the three percentages of done, todo and deprecated docs

Definition at line 117 of file doc.php.

2.2.2.6 count_proportion_doc()

```
count_proportion_doc (
    $projectId )
```

Compute proportion of each type of doc (done, to do, deprecated)

Parameters

<i>project↔ id</i>	The id of the project we want to count doc from
------------------------	---

Returns

counts Return an array containing the total number of docs, and the numbers of the done, to do, deprecated docs

Definition at line 74 of file doc.php.

2.2.2.7 delete_doc()

```
delete_doc (
    $id,
    $state )
```

Delete a doc element with its id.

Parameters

<i>id</i>	The id of the doc we want to delete
<i>state</i>	The state of the doc that will be used only in the return instruction

Returns

state Return the state of the deleted doc. This way the jQuery code will know what section of the page that needs to be refreshed

Definition at line 243 of file doc.php.

2.2.2.8 edit_doc()

```
edit_doc (
    $id,
    $name,
    $description,
    $state )
```

Modify the informations of a doc from its id.

Parameters

<i>id</i>	The id of the doc we want to modify
<i>name</i>	The new name of the doc element (can be the same as the old one)
<i>description</i>	The new description of the doc (can be the same as the old one)
<i>state</i>	The new state of the doc (can be the same as the old one)

Returns

state Return the state of the edited doc. This way the jQuery code will know what section of the page that needs to be refreshed

Definition at line 216 of file doc.php.

2.2.2.9 get_all_deprecated_doc()

```
get_all_deprecated_doc (
    $projectId )
```

Get all the doc that is marked as 'deprecated' in the database.

Parameters

<i>projectId</i>	The id of the project we want to get the deprecated doc from
------------------	--

Returns

docs All the deprecated doc found for this project

Definition at line 54 of file doc.php.

2.2.2.10 get_all_done_doc()

```
get_all_done_doc (
    $projectId )
```

Get all the doc that is marked as 'done' in the database.

Parameters

<i>projectId</i>	The id of the project we want to get the done doc from
------------------	--

Returns

docs All the done doc found for this project

Definition at line 14 of file doc.php.

2.2.2.11 get_all_todo_doc()

```
get_all_todo_doc (
    $projectId )
```

Get all the doc that is marked as 'todo' in the database.

Parameters

<i>projectId</i>	The id of the project we want to get the to do doc from
------------------	---

Returns

docs All the to do doc found for this project

Definition at line 34 of file doc.php.

2.3 /autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/faq.php File Reference

Various functions to communicate with the faq and faq_comments table of the database. Essentially CRUD functions.

Functions

- [getIdCategory](#) (\$nameCat)
This function get the id of a faq category from its name.
- [getQA](#) (\$category)
This function get all the couples questions/answers belonging to a given faq category.
- [editQA](#) (\$id)
Modify the informations of a question/answer element from its id (given by a POST request) For administrators only.
- [addQA](#) ()
Add a new question/answer element to the faq table The q/a is get from a POST request For administrators only.
- [delQA](#) (\$id)
Delete a question/answer element from the faq table from its id For administrators only.
- [searchQA](#) (\$keywords)
Get all question/answer elements from the faq table that match the given keyword.
- [addCategory](#) ()
Add a new faq category to the database The category is get by a POST request For administrators only.
- [delCategory](#) ()
Delete a faq category (and all the question/answer elements it contains by using sql cascades) The category is get by POST request For administrators only.
- [getCategories](#) ()
Get all the faq catagories that exist in the database.

2.3.1 Detailed Description

Various functions to communicate with the `faq` and `faq_comments` table of the database. Essentially CRUD functions.

Faq

2.3.2 Function Documentation

2.3.2.1 `delQA()`

```
delQA (
    $id )
```

Delete a question/answer element from the `faq` table from its id For administrators only.

Parameters

<i>id</i>	The id of the question/answer element we want to delete
-----------	---

Definition at line 122 of file `faq.php`.

2.3.2.2 `editQA()`

```
editQA (
    $id )
```

Modify the informations of a question/answer element from its id (given by a POST request) For administrators only.

Parameters

<i>id</i>	The id of the element from the <code>faq</code> table
-----------	---

Definition at line 62 of file `faq.php`.

2.3.2.3 `getCategories()`

```
getCategories ( )
```

Get all the `faq` catagories that exist in the database.

Returns

categories All the categories found in the database

Definition at line 208 of file faq.php.

2.3.2.4 getIdCategory()

```
getIdCategory (
    $nameCat )
```

This function get the id of a faq category from its name.

Parameters

<i>nameCat</i>	The name of the faq category
----------------	------------------------------

Returns

id The id of the category specified by its name

Definition at line 15 of file faq.php.

2.3.2.5 getQA()

```
getQA (
    $category )
```

This function get all the couples questions/answers belonging to a given faq category.

Parameters

<i>category</i>	The faq category we want the q/a from
-----------------	---------------------------------------

Returns

qa All the elements in the faq table corresponding the the specified category

Definition at line 36 of file faq.php.

2.3.2.6 searchQA()

```
searchQA (
    $keywords )
```

Get all question/answer elements from the faq table that match the given keyword.

Parameters

<i>keywords</i>	The keywords that will be searched in the database
-----------------	--

Returns

qa All the question/answer elements that match the request

Definition at line 143 of file faq.php.

2.4 /autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/logout.php

File Reference

This file contains only one function that allows you to disconnect from the website.

Functions

- [logout](#) ()
This function removes session variables to disconnect the user .

2.4.1 Detailed Description

This file contains only one function that allows you to disconnect from the website.

logout

2.5 /autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/modPanel.php

File Reference

Various functions to enable administrator monitoring from the mod panel page. Mainly CRUD functions.

Functions

- [getAllUsers](#) (\$role)
Get all the users in the database from a specified role.
- [changeUserRole](#) (\$user, \$newrole)
Change the role of a given user to match the new given role Only possible in the case of an administrator editing the role of a regular user.
- [deleteUser](#) (\$user)
Delete a given user from its username Only possible in the case of an administrator deleting a regular user.

2.5.1 Detailed Description

Various functions to enable administrator monitoring from the mod panel page. Mainly CRUD functions.

Mod Panel

2.5.2 Function Documentation

2.5.2.1 changeUserRole()

```
changeUserRole (
    $user,
    $newrole )
```

Change the role of a given user to match the new given role Only possible in the case of an administrator editing the role of a regular user.

Parameters

<i>user</i>	The username of the user that will change role
<i>newrole</i>	The new role that will be given to this user

Definition at line 39 of file modPanel.php.

2.5.2.2 deleteUser()

```
deleteUser (
    $user )
```

Delete a given user from its username Only possible in the case of an administrator deleting a regular user.

Parameters

<i>user</i>	The username of the user that will be deleted
-------------	---

Definition at line 79 of file modPanel.php.

2.5.2.3 getAllUsers()

```
getAllUsers (
    $role )
```

Get all the users in the database from a specified role.

Parameters

<i>role</i>	The role we want to get all the users from
-------------	--

Returns

users All the users that have the specified role

Definition at line 15 of file modPanel.php.

2.6 /autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/profile.php File Reference

Various functions, who have for objectif to give all the necessary for the controller profile you will find fonctions for project_invitation user essentially CRUD function.

Functions

- [getUserProfile](#) (\$userName)
This function return the information about the given user.
- [getUserNbParticipation](#) (\$userId)
This function Return the number of projects in which a user participates.
- [getUserInvitationsOrRequest](#) (\$isRequest, \$userId)
Returns all invitations or requests for a given user.
- [CancelRequest](#) (\$projectIdToCancel, \$userId)
This function allows you to cancel a request by deleting the line in the database. this function is automaticly call when you accept an invitation.
- [AcceptInvitaion](#) (\$projectIdToJoin, \$userId)
This function adds a user who has received an invitation to a project to the project.
- [deleteAccount](#) (\$userName)
This function delete the user with his username.
- [editInfo](#) (\$user)

2.6.1 Detailed Description

Various functions, who have for objectif to give all the necessary for the controller profile you will find fonctions for project_invitation user essentially CRUD function.

profile

2.6.2 Function Documentation**2.6.2.1 AcceptInvitaion()**

```
AcceptInvitaion (
    $projectIdToJoin,
    $userId )
```

This function adds a user who has received an invitation to a project to the project.

Parameters

<i>projectIdToJoin</i>	the project id
<i>userId</i>	the user id

Returns

1 if succes -1 if an exception occurs

Definition at line 117 of file profile.php.

2.6.2.2 CancelRequest()

```
CancelRequest (
    $projectIdToCancel,
    $userId )
```

This function allows you to cancel a request by deleting the line in the database. this function is automaticly call when you accept an invitation.

Parameters

<i>projectIdToCancel</i>	the project id
<i>userId</i>	the user id

Returns

1 if succes -1 if an exception occurs

Definition at line 89 of file profile.php.

2.6.2.3 deleteAccount()

```
deleteAccount (
    $userName )
```

This function delete the user with his username.

Parameters

<i>userName</i>	the user name of the usert that you want to remove
-----------------	--

Returns

1 if succes -1 if an exception occurs

Definition at line 145 of file profile.php.

2.6.2.4 editInfo()

```
editInfo (
    $user )
```

Update the informations of the user specified by his username from a POST request

Definition at line 166 of file profile.php.

2.6.2.5 getUserInvitationsOrRequest()

```
getUserInvitationsOrRequest (
    $isRequest,
    $userId )
```

Returns all invitations or requests for a given user.

Parameters

<i>isRequest</i>	1 for a request and 0 for an invitation
<i>userId</i>	The id of the user that you want the number of project

Returns

the PDOStatement contening thenumber of project about given user return -1 if an exception occurs

Definition at line 64 of file profile.php.

2.6.2.6 getUserNbParticipation()

```
getUserNbParticipation (
    $userId )
```

This function Return the number of projects in which a user participates.

Parameters

<i>userId</i>	The id of the user that you want the number of project
---------------	--

Returns

the number of project about given user return -1 if an exception occurs

Definition at line 40 of file profile.php.

2.6.2.7 getUserProfile()

```
getUserProfile (
    $userName )
```

This function return the information about the given user.

Parameters

<i>userName</i>	The name of the user that you want the information
-----------------	--

Returns

an array contening the information about given user return -1 if an exception occurs

Definition at line 15 of file profile.php.

2.7 /autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/projects.php

File Reference

Various functions, who have for objectif to give all the necessary for the controller projects you will find fonctions for project user inside project essentially CRUD function.

Functions

- [get_all_project_by_user_id](#) (\$id)
This function return all projects of which a user is a member.
- [get_all_project_without_user_id](#) (\$id)
This function retrieves all projects to which a user is not a member.
- [remove_by_project_id](#) (\$id)
this function remove a given project inside database on project table
- [create_new_project](#) (\$userId, \$name, \$description, \$visibility=1)
This function creates a new project and inserts it into the project table.

- [associat_project_and_user](#) (\$userId, \$projectId, \$role)

This function links a user to a project, this function is automatically called to create a project to associate the creator of the project to the project as project manager.

- [leave_a_project](#) (\$userId, \$projectId)

This function removes a user from a project.

- [add_invitation_request](#) (\$requesterUserId, \$projectId)

This function adds an invitation to a project to a user.

- [check_if_request_already_exist](#) (\$requesterUserId, \$projectId)

This function checks if an invitation already exists this function is called when adding an invitation request.

2.7.1 Detailed Description

Various functions, who have for objectif to give all the necessary for the controller projects you will find fonctions for project user inside project essentially CRUD function.

projects

2.7.2 Function Documentation

2.7.2.1 add_invitation_request()

```
add_invitation_request (
    $requesterUserId,
    $projectId )
```

This function adds an invitation to a project to a user.

Parameters

<i>requester↔ UserId</i>	The identifier of the user that you want to invite to the project
<i>projectId</i>	The identifier of the project that you want to invite the user.

Returns

1 if succes -1 if an exception occurs

Definition at line 183 of file projects.php.

2.7.2.2 associat_project_and_user()

```
associat_project_and_user (
    $userId,
```

```

    $projectId,
    $role )

```

This function links a user to a project, this function is automatically called to create a project to associate the creator of the project to the project as project manager.

Parameters

<i>userId</i>	The identifier of the user that you want to link to the project
<i>projectId</i>	The identifier of the project that you want to link the user.
<i>role</i>	The role of the user you want to link to the project.

Returns

1 if succes -1 if an exception occurs

Definition at line 126 of file projects.php.

2.7.2.3 check_if_request_already_exist()

```

check_if_request_already_exist (
    $requesterUserId,
    $projectId )

```

This function checks if an invitation already exists this function is called when adding an invitation request.

Parameters

<i>requesterUserId</i>	The identifier of the user that you want to invite to the project
<i>projectId</i>	The identifier of the project that you want to invite the user.

Returns

1 if succes -1 if an exception occurs

Definition at line 216 of file projects.php.

2.7.2.4 create_new_project()

```

create_new_project (
    $userId,
    $name,
    $description,
    $visibility = 1 )

```

This function creates a new project and inserts it into the project table.

Parameters

<i>userId</i>	The identifier of the user who created the project this user will be declared by default head of this project
<i>name</i>	The name of the project you want to create.
<i>description</i>	The description of the project you want to create.
<i>visibility</i>	The visibility of the project you want to create by default the project is public.

Returns

The last insert index on the database or -1 if an exception occurs

Definition at line 95 of file projects.php.

2.7.2.5 get_all_project_by_user_id()

```
get_all_project_by_user_id (
    $id )
```

This function return all projects of which a user is a member.

Parameters

<i>id</i>	The identifier of the user for whom you want to retrieve all projects
-----------	---

Returns

the PDOStatement contening all the project return -1 if an exception occurs

Definition at line 16 of file projects.php.

2.7.2.6 get_all_project_without_user_id()

```
get_all_project_without_user_id (
    $id )
```

This function retrieves all projects to which a user is not a member.

Parameters

<i>id</i>	The identifier of the user for whom you want to retrieve all projects
-----------	---

Returns

the PDOStatement contening all the project return -1 if an exception occurs

Definition at line 39 of file projects.php.

2.7.2.7 leave_a_project()

```
leave_a_project (
    $userId,
    $projectId )
```

This function removes a user from a project.

Parameters

<i>userId</i>	The identifier of the user that you want to unlink to the project
<i>projectId</i>	The identifier of the project that you want to unlink the user.

Returns

1 if succes -1 if an exception occurs

Definition at line 155 of file projects.php.

2.7.2.8 remove_by_project_id()

```
remove_by_project_id (
    $id )
```

this function remove a given project inside database on project table

Parameters

<i>id</i>	the project id that you want to remove
-----------	--

Returns

1 if succes -1 if an exception occurs

Definition at line 62 of file projects.php.

2.8 /autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/release.php File Reference

Various functions, who have for objectif to give all the necessary for the controller release you will find fonctions for project_commits essentially CRUD function.

Functions

- [get_git_url](#) (\$projectID)
This function return the git url of a given project.
- [save_all_commit](#) (\$projectID, \$allCommits)
save all given commits int the database in project_commit table
- [deprecate_all_test](#) (\$projectID)
This function is automatically called when you insert a new commit in the database and makes all the tests of a project obsolete.
- [change_git_url](#) (\$projectID, \$gitUrl)
This function allows you to modify the git URL associated with a project.
- [delete_all_commits](#) (\$projectID)
This function removes all commits for a project from the database. This function is automatically called when you change the URL of a project's git.
- [get_all_commit](#) (\$projectID)
This function returns all the commits for a given project.
- [get_last_commit](#) (\$projectID)
This function returns the last commit for a given project.

2.8.1 Detailed Description

Various functions, who have for objectif to give all the necessary for the controller release you will find fonctions for project_commits essentially CRUD function.

release

2.8.2 Function Documentation

2.8.2.1 change_git_url()

```
change_git_url (
    $projectID,
    $gitUrl )
```

This function allows you to modify the git URL associated with a project.

Parameters

<i>projectID</i>	The identifier of the project
<i>gitUrl</i>	the new gitUrl

Returns

1 if succes -1 if an exception occurs

Definition at line 101 of file release.php.

2.8.2.2 delete_all_commits()

```
delete_all_commits (
    $projectID )
```

This function removes all commits for a project from the database. This function is automatically called when you change the URL of a project's git.

Parameters

<i>projectID</i>	The identifier of the project
------------------	-------------------------------

Returns

1 if succes -1 if an exception occurs

Definition at line 128 of file release.php.

2.8.2.3 deprecate_all_test()

```
deprecate_all_test (
    $projectID )
```

This function is automatically called when you insert a new commit in the database and makes all the tests of a project obsolete.

Parameters

<i>projectID</i>	The identifier of the project
------------------	-------------------------------

Returns

1 if succes -1 if an exception occurs

Definition at line 80 of file release.php.

2.8.2.4 get_all_commit()

```
get_all_commit (
    $projectID )
```

This function returns all the commits for a given project.

Parameters

<i>projectID</i>	The identifier of the project that you the commits
------------------	--

Returns

the PDOStatement contening the all commit or return -1 if an exception occurs

Definition at line 153 of file release.php.

2.8.2.5 get_git_url()

```
get_git_url (
    $projectID )
```

This function return the git url of a given project.

Parameters

<i>projectID</i>	The identifier of the project that you want the url
------------------	---

Returns

the PDOStatement contening the gitUrl return -1 if an exception occurs

Definition at line 16 of file release.php.

2.8.2.6 get_last_commit()

```
get_last_commit (
    $projectID )
```

This function returns the last commit for a given project.

Parameters

<i>projectID</i>	The identifier of the project that you the commit
------------------	---

Returns

the PDOStatement contening the commit or return -1 if an exception occurs

Definition at line 178 of file release.php.

2.8.2.7 save_all_commit()

```
save_all_commit (
    $projectID,
    $allCommits )
```

save all given commits int the database in project_commit table

Parameters

<i>projectID</i>	The identifier of the project that you to save the commit
<i>allCommits</i>	An array containing the commits to be inserted must be a two-dimensional array

Returns

1 if succes -1 if an exception occurs

Definition at line 39 of file release.php.

2.9 /autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/selectedProject.php File Reference ↩

Various functions, who have for objectif to give all the necessary for selectedProject you will find fonctions for project user essentially CRUD function.

Functions

- [get_project_master](#) (\$id)
This function returns the leader of a given project.
- [get_all_project_members](#) (\$id)
This function returns all the member of the project.
- [get_all_project_joining_requests](#) (\$id)
This function returns all the request for the given project.
- [get_all_project_invitations](#) (\$id)
This function returns all the sendend invitation for the given project.
- [get_project_by_id](#) (\$id)
This function returns all the informations for the given project.
- **editProject** (\$id)
- [acceptRequest](#) (\$project_id, \$user_id, \$role)
This function adds a user who has send a request to a project to the project.

- `deleteMember` (\$project_id, \$user_id)
This function deletes a given user from a given project.
- `is_master` (\$id, \$project_id)
This function allows you to know if a given user is the project manager.
- `is_member` (\$id, \$project_id)
This function allows you to know if a given user is a project member.
- `deleteInvitationOrRequest` (\$project_id, \$user_id)
Removes the invitation or query for a given project and user from the project_invitation table.

2.9.1 Detailed Description

Various functions, who have for objectif to give all the necessary for selectedProject you will find fonctions for project user essentially CRUD function.

selectedProject

2.9.2 Function Documentation

2.9.2.1 acceptRequest()

```
acceptRequest (
    $project_id,
    $user_id,
    $role )
```

This function adds a user who has send a request to a project to the project.

Parameters

<i>project_id</i>	the project id
<i>user_id</i>	the user id
<i>role</i>	the role of the user in the project id

Returns

1 if succes -1 if an exception occurs

Definition at line 142 of file selectedProject.php.

2.9.2.2 deleteInvitationOrRequest()

```
deleteInvitationOrRequest (
    $project_id,
    $user_id )
```

Removes the invitation or query for a given project and user from the `project_invitation` table.

Parameters

<i>project_id</i>	the project id
<i>user_id</i>	the user id

Returns

1 if succes -1 if an exception occurs

Definition at line 261 of file selectedProject.php.

2.9.2.3 deleteMember()

```
deleteMember (
    $project_id,
    $user_id )
```

This function deletes a given user from a given project.

Parameters

<i>project_id</i>	the project id
<i>user_id</i>	the user id

Returns

1 if succes -1 if an exception occurs

Definition at line 173 of file selectedProject.php.

2.9.2.4 get_all_project_invitations()

```
get_all_project_invitations (
    $id )
```

This function returns all the sendend invitation for the given project.

Parameters

<i>id</i>	the project id
-----------	----------------

Returns

The PDOStatement contening user and project_invitation of the given project or -1 if an exception occurs

Definition at line 77 of file selectedProject.php.

2.9.2.5 get_all_project_joining_requests()

```
get_all_project_joining_requests (
    $id )
```

This function returns all the request for the given project.

Parameters

<i>id</i>	the project id
-----------	----------------

Returns

The PDOStatement contening user and project_invitation of the given project or -1 if an exception occurs

Definition at line 56 of file selectedProject.php.

2.9.2.6 get_all_project_members()

```
get_all_project_members (
    $id )
```

This function returns all the member of the project.

Parameters

<i>id</i>	the project id
-----------	----------------

Returns

The PDOStatement contening all the member of the given project or -1 if an exception occurs

Definition at line 35 of file selectedProject.php.

2.9.2.7 get_project_by_id()

```
get_project_by_id (
    $id )
```

This function returns all the informations for the given project.

Parameters

<i>id</i>	the project id
-----------	----------------

Returns

The PDOStatement contening all information of the given project or -1 if an exception occurs

Definition at line 98 of file selectedProject.php.

2.9.2.8 get_project_master()

```
get_project_master (
    $id )
```

This function returns the leader of a given project.

Parameters

<i>id</i>	the project id
-----------	----------------

Returns

The PDOStatement contening the project master or -1 if an exception occurs

Definition at line 15 of file selectedProject.php.

2.9.2.9 is_master()

```
is_master (
    $id,
    $project_id )
```

This function allows you to know if a given user is the project manager.

Parameters

<i>project_id</i>	the project id
<i>user_id</i>	the user id

Returns

true if is master false if not and -1 if an exception occurs

Definition at line 194 of file selectedProject.php.

2.9.2.10 is_member()

```
is_member (
    $id,
    $project_id )
```

This function allows you to know if a given user is a project member.

Parameters

<i>project_id</i>	the project id
<i>user_id</i>	the user id

Returns

true if is member false if not and -1 if an exception occurs

Definition at line 227 of file selectedProject.php.

2.10 /autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/sprints.php File Reference

Various functions, who have for objectif to give all the necessary for sprint controller you will find fonctions for project user essentially CRUD function.

Functions

- [get_all_sprints](#) (\$projectId)
This function return a list of all sprint with extra information(the number of us and the effort) of a given project.
- [add_sprint_background](#) (\$startDate, \$endDate)
This function returns the bootstrap code of the background of a sprint according to its start date and end date.
- [create_new_sprint](#) (\$name, \$start, \$end, \$projectId)
This function returns the bootstrap code of the background of a sprint according to its start date and end date.
- [delete_sprint_by_id](#) (\$sprintId)
This function removes a sprint from the sprint table according to its identifier.
- [get_all_project_members_and_master](#) (\$id)
This function returns all members of a project including the project leader.
- [create_new_task](#) (\$name, \$description, \$dod, \$predecessor, \$time, \$sprintId, \$memberId)
This function allows you to create a task in a given sprint.Insert a line in the task table.
- [update_task](#) (\$id, \$name, \$description, \$dod, \$predecessor, \$time, \$memberId)
This function allows you to modify a given task.
- [get_all_task_inside_sprint](#) (\$sprintId)

This function allows you to return the complete list of tasks inside a sprint.

- [get_all_us_inside_task](#) (\$taskId)

This function return the us linked to a task.

- [unlink_us_from_task](#) (\$taskId, \$allUSToUnlink)

This function unlink USs to a given Task by deleting from the table inside_sprint_task_us.

- [link_us_to_task](#) (\$taskId, \$allUSToLink)

This function USs to a specified task.

- [get_all_us_inside_sprint](#) (\$sprintId)

This function allows you to return the complete list of userstory inside a sprint.

- [get_all_us_inside_project](#) (\$projectId)

This function allows you to return the complete list of userstory inside a project.

- [switch_task_state](#) (\$taskId, \$state)

This function allows you to change the status of a task.

- [remove_task](#) (\$taskId)

This function remove a task.

- [link_us_to_sprint](#) (\$sprintId, \$allUSToLink)

This function allows you to link the us to a sprint.

- [unlink_us_from_sprint](#) (\$sprintId, \$allUSToUnlink)

This function allows you to unlink the us to a sprint.

- [count_nb_task_by_state_in_sprint](#) (\$sprintId)

This method returns an array of three columns with the number of tasks.

2.10.1 Detailed Description

Various functions, who have for objectif to give all the necessary for sprint controller you will find fonctions for project user essentially CRUD function.

sprints

2.10.2 Function Documentation

2.10.2.1 add_sprint_background()

```
add_sprint_background (
    $startDate,
    $endDate )
```

This function returns the bootstrap code of the background of a sprint according to its start date and end date.

Parameters

<i>startDate</i>	the startDate of the sprint
<i>endDate</i>	the endDate of the sprint

Returns

a string contening the html bootstrap code for the background color

Definition at line 56 of file sprints.php.

2.10.2.2 count_nb_task_by_state_in_sprint()

```
count_nb_task_by_state_in_sprint (
    $sprintId )
```

This method returns an array of three columns with the number of tasks.

Parameters

<i>sprintId</i>	the sprint id
-----------------	---------------

Returns

an array with the task bomb to the number of tage onGoin and the number of task done -1 if an exception occurs

Definition at line 488 of file sprints.php.

2.10.2.3 create_new_sprint()

```
create_new_sprint (
    $name,
    $start,
    $end,
    $projectId )
```

This function returns the bootstrap code of the background of a sprint according to its start date and end date.

Parameters

<i>startDate</i>	the startDate of the sprint
<i>endDate</i>	the endDate of the sprint

Returns

a string contening the html bootstrap code for the background color

Definition at line 77 of file sprints.php.

2.10.2.4 create_new_task()

```
create_new_task (
    $name,
    $description,
    $dod,
    $predecessor,
    $time,
    $sprintId,
    $memberId )
```

This function allows you to create a task in a given sprint. Insert a line in the task table.

Parameters

<i>name</i>	the task name
<i>description</i>	the description of the task
<i>dod</i>	the definition of done of the task
<i>predecessor</i>	the predecessor task list of the task
<i>time</i>	the time for finish the task
<i>sprintId</i>	the sprint where to add the task
<i>memberId</i>	the member that is responsible to do the task

Returns

The id of the insertion on the table or -1 if an exception occurs

Definition at line 154 of file sprints.php.

2.10.2.5 delete_sprint_by_id()

```
delete_sprint_by_id (
    $sprintId )
```

This function removes a sprint from the sprint table according to its identifier.

Parameters

<i>sprintId</i>	the identifier of the sprint to be deleted
-----------------	--

Returns

return 1 if success or -1 if an exception occurs

Definition at line 105 of file sprints.php.

2.10.2.6 `get_all_project_members_and_master()`

```
get_all_project_members_and_master (
    $id )
```

This function returns all members of a project including the project leader.

Parameters

<i>id</i>	the project id
-----------	----------------

Returns

The PDOStatement contening all the member and master of a project or -1 if an exception occurs

Definition at line 128 of file sprints.php.

2.10.2.7 `get_all_sprints()`

```
get_all_sprints (
    $projectId )
```

This function return a list of all sprint with extra information(the number of us and the effort) of a given project.

Parameters

<i>id</i>	the project id
-----------	----------------

Returns

The PDOStatement contening all the sprints or -1 if an exception occurs

Definition at line 15 of file sprints.php.

2.10.2.8 `get_all_task_inside_sprint()`

```
get_all_task_inside_sprint (
    $sprintId )
```

This function allows you to return the complete list of tasks inside a sprint.

Parameters

<i>sprint↵ id</i>	the sprint id
-----------------------	---------------

Returns

The PDOStatement contening all the task of a sprint or -1 if an exception occurs

Definition at line 223 of file sprints.php.

2.10.2.9 get_all_us_inside_project()

```
get_all_us_inside_project (
    $projectId )
```

This function allows you to return the complete list of userstory inside a project.

Parameters

<i>projectId</i>	the project id
------------------	----------------

Returns

The PDOStatement contening all the userstory of a project or -1 if an exception occurs

Definition at line 368 of file sprints.php.

2.10.2.10 get_all_us_inside_sprint()

```
get_all_us_inside_sprint (
    $sprintId )
```

This function allows you to return the complete list of userstory inside a sprint.

Parameters

<i>sprintId</i>	the sprint id
-----------------	---------------

Returns

The PDOStatement contening all the userstory of a sprint or -1 if an exception occurs

Definition at line 330 of file sprints.php.

2.10.2.11 `get_all_us_inside_task()`

```
get_all_us_inside_task (
    $taskId )
```

This function return the us linked to a task.

Parameters

<i>taskId</i>	The task id
---------------	-------------

Returns

The PDOStatement contening the USs linked to the task

Definition at line 244 of file sprints.php.

2.10.2.12 `link_us_to_sprint()`

```
link_us_to_sprint (
    $sprintId,
    $allUSToLink )
```

This function allows you to link the us to a sprint.

Parameters

<i>sprintId</i>	the sprint id
<i>allUSToLink</i>	an array contening all the us to link

Returns

return 1 if success or -1 if an exception occurs

Definition at line 435 of file sprints.php.

2.10.2.13 `link_us_to_task()`

```
link_us_to_task (
    $taskId,
    $allUSToLink )
```

This function USs to a specified task.

Parameters

<i>taskId</i>	The task id
<i>allUSToLink</i>	Array composed of USs

Returns

Return -1 if an error occurs or 1 if it pass

Definition at line 293 of file sprints.php.

2.10.2.14 remove_task()

```
remove_task (
    $taskId )
```

This function remove a task.

Parameters

<i>task↔ Id</i>	the task id
---------------------	-------------

Returns

return 1 if success or -1 if an exception occurs

Definition at line 414 of file sprints.php.

2.10.2.15 switch_task_state()

```
switch_task_state (
    $taskId,
    $state )
```

This function allows you to change the status of a task.

Parameters

<i>task↔ Id</i>	the task id
<i>state</i>	the new state of a task possible value : 'todo', 'onGoing', 'done'

Returns

return 1 if success or -1 if an exception occurs

Definition at line 390 of file sprints.php.

2.10.2.16 unlink_us_from_sprint()

```
unlink_us_from_sprint (
    $sprintId,
    $allUSToUnlink )
```

This function allows you to unlink the us to a sprint.

Parameters

<i>taskId</i>	the task id
<i>allUSToLink</i>	an array contening all the us to unlink

Returns

return 1 if success or -1 if an exception occurs

Definition at line 462 of file sprints.php.

2.10.2.17 unlink_us_from_task()

```
unlink_us_from_task (
    $taskId,
    $allUSToUnlink )
```

This function unlink USs to a given Task by deleting from the table inside_sprint_task_us.

Parameters

<i>taskId</i>	the task id
<i>allUSToUnlink</i>	Array composed of USs

Returns

Return -1 if an error occurs or 1 if it pass

Definition at line 266 of file sprints.php.

2.10.2.18 update_task()

```
update_task (
    $id,
    $name,
    $description,
    $dod,
    $predecessor,
    $time,
    $memberId )
```

This function allows you to modify a given task.

Parameters

<i>name</i>	the new task name
<i>description</i>	the new description of the task
<i>dod</i>	the new definition of done of the task
<i>predecessor</i>	the new predecessor task list of the task
<i>time</i>	the new time for finish the task
<i>sprintId</i>	the new sprint where to add the task
<i>memberId</i>	the new member that is responsible to do the task

Returns

The id of the modified task on the table or -1 if an exception occurs

Definition at line 193 of file sprints.php.

2.11 /autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/tests.php File Reference

Various functions to make the tests page works Mainly CRUD functions, plus some to deal with the progression bar.

Functions

- [get_all_passed_tests](#) (\$projectId)
Get all the tests that are marked as 'passed' in the database.
- [get_all_deprecated_tests](#) (\$projectId)
Get all the tests that are marked as 'deprecated' in the database.
- [get_all_failed_tests](#) (\$projectId)
Get all the tests that are marked as 'failed' in the database.
- [get_all_never_run_tests](#) (\$projectId)
Get all the tests that are marked as 'never run' in the database.
- [count_proportion](#) (\$projectId)
Compute proportion of each type of test (passed, failed, deprecated, never run)
- [compute_proportion](#) (\$projectId)
Compute the tests proportions of each category (passed, failed, deprecated, never run) to make percentages.

- `add_new_test` (\$projectId, \$name, \$description, \$state)
Function to add a new test to the specified projet.
- `pass_all_tests` (\$projectId)
Function to mark all tests of a given project as passed and refresh their execution date.
- `edit_test` (\$id, \$name, \$description, \$state)
Modify the informations of a test from its id.
- `delete_test` (\$id, \$state)
Delete a test from its id.
- `change_state` (\$id, \$state)
Change the state of a test from its id and refresh its execution date. The function can be used only to refresh last execution date (without state change)
- `check_deprecated` (\$projectId)
This function check all passed tests in the database for a given project and mark them as deprecated if they are too old.

2.11.1 Detailed Description

Various functions to make the tests page works Mainly CRUD functions, plus some to deal with the progression bar.

Tests

2.11.2 Function Documentation

2.11.2.1 add_new_test()

```
add_new_test (
    $projectId,
    $name,
    $description,
    $state )
```

Function to add a new test to the specified projet.

Parameters

<i>projectId</i>	The id of the project we want to add the test to
<i>name</i>	The name of the test
<i>description</i>	The text description of the test
<i>state</i>	The enum state of the test (can be passed, failed, deprecated or never_run)

Returns

state Return the state of the added test (this state will be used by the jQuery code to know which part of the page to refresh)

Definition at line 184 of file tests.php.

2.11.2.2 change_state()

```
change_state (
    $id,
    $state )
```

Change the state of a test from its id and refresh its execution date. The function can be used only to refresh last execution date (without state change)

Parameters

<i>id</i>	The id of the test that will change state
<i>state</i>	The new state of the test

Returns

stmt The returned statement of the executed sql request

Definition at line 293 of file tests.php.

2.11.2.3 check_deprecated()

```
check_deprecated (
    $projectId )
```

This function check all passed tests in the database for a given project and mark them as deprecated if they are too old.

Parameters

<i>projectId</i>	The id of the project we want to check deprecated tests from
------------------	--

Returns

number Return the number of deprecated tests found. If there was an error during execution, return -1

Definition at line 320 of file tests.php.

2.11.2.4 compute_proportion()

```
compute_proportion (
    $projectId )
```

Compute the tests proportions of each category (passed, failed, deprecated, never run) to make percentages.

Parameters

<i>project↔ Id</i>	The id of the project we want to compute the tests proportions from
------------------------	---

Returns

percentages Return an array containing the four percentages of passed, failed, deprecated and never run tests

Definition at line 141 of file tests.php.

2.11.2.5 count_proportion()

```
count_proportion (
    $projectId )
```

Compute proportion of each type of test (passed, failed, deprecated, never run)

Parameters

<i>project↔ Id</i>	The id of the project we want to count tests from
------------------------	---

Returns

counts Return an array containing the total number of tests, and the numbers of the passed, failed, deprecated and never run tests

Definition at line 94 of file tests.php.

2.11.2.6 delete_test()

```
delete_test (
    $id,
    $state )
```

Delete a test from its id.

Parameters

<i>id</i>	The id of the test we want to delete
<i>state</i>	The state of the test that will be used only in the return instruction

Returns

state Return the state of the deleted test. This way the jQuery code will know what section of the page that needs to be refreshed

Definition at line 270 of file tests.php.

2.11.2.7 edit_test()

```
edit_test (
    $id,
    $name,
    $description,
    $state )
```

Modify the informations of a test from its id.

Parameters

<i>id</i>	The id of the test we want to modify
<i>name</i>	The new name of the test (can be the same as the old one)
<i>description</i>	The new description of the test (can be the same as the old one)
<i>state</i>	The new state of the test (can be the same as the old one)

Returns

state Return the state of the edited test. This way the jQuery code will know what section of the page that needs to be refreshed

Definition at line 243 of file tests.php.

2.11.2.8 get_all_deprecated_tests()

```
get_all_deprecated_tests (
    $projectId )
```

Get all the tests that are marked as 'deprecated' in the database.

Parameters

<i>projectId</i>	The id of the project we want to get the deprecated tests from
------------------	--

Returns

tests All the deprecated tests found for this project

Definition at line 34 of file tests.php.

2.11.2.9 `get_all_failed_tests()`

```
get_all_failed_tests (
    $projectId )
```

Get all the tests that are marked as 'failed' in the database.

Parameters

<i>projectId</i>	The id of the project we want to get the failed tests from
------------------	--

Returns

tests All the failed tests found for this project

Definition at line 54 of file tests.php.

2.11.2.10 `get_all_never_run_tests()`

```
get_all_never_run_tests (
    $projectId )
```

Get all the tests that are marked as 'never run' in the database.

Parameters

<i>projectId</i>	The id of the project we want to get the never run tests from
------------------	---

Returns

tests All the never run tests found for this project

Definition at line 74 of file tests.php.

2.11.2.11 `get_all_passed_tests()`

```
get_all_passed_tests (
    $projectId )
```

Get all the tests that are marked as 'passed' in the database.

Parameters

<i>projectId</i>	The id of the project we want to get the passed tests from
------------------	--

Returns

tests All the passed tests found for this project

Definition at line 14 of file tests.php.

2.11.2.12 pass_all_tests()

```
pass_all_tests (
    $projectId )
```

Function to mark all tests of a given project as passed and refresh their execution date.

Parameters

<i>projectId</i>	The id of the project that will have all its tests marked as passed
------------------	---

Returns

stmt The statement value of the executed sql instruction

Definition at line 213 of file tests.php.

2.12 /autofs/unityaccount/pages_web_perso/msjuan/gr1_eq2_Dev/src/model/utility.php File Reference

Various functions, who have for objectif to give all the necessary for all controller you will find fonctions for project user essentially CRUD function.

Functions

- [get_all_user_not_in_project](#) (\$projectId)
This function returns all users who do not belong to a project.
- [add_invitation_by_user_name](#) (\$requesterUserName, \$projectId)
This function adds an invitation to the database and inserts a line in the project_invitation table.
- [check_if_request_already_exist](#) (\$requesterUserName, \$projectId, \$bdd)
This function checks if an invitation already exists this function is called when adding an invitation request.

2.12.1 Detailed Description

Various functions, who have for objectif to give all the necessary for all controller you will find fonctions for project user essentially CRUD function.

profile

2.12.2 Function Documentation

2.12.2.1 add_invitation_by_user_name()

```
add_invitation_by_user_name (
    $requesterUserName,
    $projectId )
```

This function adds an invitation to the database and inserts a line in the project_invitation table.

Parameters

<i>requesterUserName</i>	the username of the user that you want to invite to the project
<i>projectId</i>	the project id

Returns

return 1 if success or -1 if an exception occurs

Definition at line 45 of file utility.php.

2.12.2.2 check_if_request_already_exist()

```
check_if_request_already_exist (
    $requesterUserName,
    $projectId,
    $bdd )
```

This function checks if an invitation already exists this function is called when adding an invitation request.

Parameters

<i>requester↔ UserId</i>	The identifier of the user that you want to invite to the project
<i>projectId</i>	The identifier of the project that you want to invite the user.

Returns

1 if succes -1 if an exception occurs

Definition at line 80 of file utility.php.

2.12.2.3 get_all_user_not_in_project()

```
get_all_user_not_in_project (
    $projectId )
```

This function returns all users who do not belong to a project.

Parameters

<i>projectId</i>	the project id
------------------	----------------

Returns

The PDOStatement contening all users or -1 if an exception occurs

Definition at line 16 of file utility.php.

Index

/autofs/unityaccount/pages_web_perso/msjuan/gr1_↵
eq2_Dev/src/model/backlog.php, 3
/autofs/unityaccount/pages_web_perso/msjuan/gr1_↵
eq2_Dev/src/model/doc.php, 7
/autofs/unityaccount/pages_web_perso/msjuan/gr1_↵
eq2_Dev/src/model/faq.php, 13
/autofs/unityaccount/pages_web_perso/msjuan/gr1_↵
eq2_Dev/src/model/logout.php, 16
/autofs/unityaccount/pages_web_perso/msjuan/gr1_↵
eq2_Dev/src/model/modPanel.php, 16
/autofs/unityaccount/pages_web_perso/msjuan/gr1_↵
eq2_Dev/src/model/profile.php, 19
/autofs/unityaccount/pages_web_perso/msjuan/gr1_↵
eq2_Dev/src/model/projects.php, 22
/autofs/unityaccount/pages_web_perso/msjuan/gr1_↵
eq2_Dev/src/model/release.php, 27
/autofs/unityaccount/pages_web_perso/msjuan/gr1_↵
eq2_Dev/src/model/selectedProject.php, 30
/autofs/unityaccount/pages_web_perso/msjuan/gr1_↵
eq2_Dev/src/model/sprints.php, 36
/autofs/unityaccount/pages_web_perso/msjuan/gr1_↵
eq2_Dev/src/model/tests.php, 45
/autofs/unityaccount/pages_web_perso/msjuan/gr1_↵
eq2_Dev/src/model/utility.php, 51

AcceptInvitaion
profile.php, 19
acceptRequest
selectedProject.php, 31
add_inside_project_US
backlog.php, 4
add_inside_project_role
backlog.php, 4
add_invitation_by_user_name
utility.php, 52
add_invitation_request
projects.php, 23
add_new_doc
doc.php, 8
add_new_test
tests.php, 46
add_sprint_background
sprints.php, 37
all_doc_done
doc.php, 9
associat_project_and_user
projects.php, 23
backlog.php
add_inside_project_US, 4

add_inside_project_role, 4
get_all_US_by_project_id, 5
get_all_inside_project_role, 5
modify_inside_project_US, 6
modify_inside_project_role, 5
remove_US_by_id, 7
remove_by_role_id, 6

CancelRequest
profile.php, 20
change_git_url
release.php, 27
change_state
tests.php, 46
change_state_doc
doc.php, 9
changeUserRole
modPanel.php, 17
check_deprecated
tests.php, 47
check_deprecated_doc
doc.php, 9
check_if_request_already_exist
projects.php, 24
utility.php, 52
compute_proportion
tests.php, 47
compute_proportion_doc
doc.php, 10
count_nb_task_by_state_in_sprint
sprints.php, 38
count_proportion
tests.php, 48
count_proportion_doc
doc.php, 10
create_new_project
projects.php, 24
create_new_sprint
sprints.php, 38
create_new_task
sprints.php, 38
delQA
faq.php, 14
delete_all_commits
release.php, 28
delete_doc
doc.php, 11
delete_sprint_by_id
sprints.php, 39

- delete_test
 - tests.php, 48
- deleteAccount
 - profile.php, 20
- deleteInvitationOrRequest
 - selectedProject.php, 31
- deleteMember
 - selectedProject.php, 33
- deleteUser
 - modPanel.php, 17
- deprecate_all_test
 - release.php, 28
- doc.php
 - add_new_doc, 8
 - all_doc_done, 9
 - change_state_doc, 9
 - check_deprecated_doc, 9
 - compute_proportion_doc, 10
 - count_proportion_doc, 10
 - delete_doc, 11
 - edit_doc, 11
 - get_all_deprecated_doc, 12
 - get_all_done_doc, 12
 - get_all_todo_doc, 12
- edit_doc
 - doc.php, 11
- edit_test
 - tests.php, 49
- editInfo
 - profile.php, 21
- editQA
 - faq.php, 14
- faq.php
 - delQA, 14
 - editQA, 14
 - getCategories, 14
 - getCategoryId, 15
 - getQA, 15
 - searchQA, 15
- get_all_US_by_project_id
 - backlog.php, 5
- get_all_commit
 - release.php, 28
- get_all_deprecated_doc
 - doc.php, 12
- get_all_deprecated_tests
 - tests.php, 49
- get_all_done_doc
 - doc.php, 12
- get_all_failed_tests
 - tests.php, 50
- get_all_inside_project_role
 - backlog.php, 5
- get_all_never_run_tests
 - tests.php, 50
- get_all_passed_tests
 - tests.php, 50
- get_all_project_by_user_id
 - projects.php, 25
- get_all_project_invitations
 - selectedProject.php, 33
- get_all_project_joining_requests
 - selectedProject.php, 34
- get_all_project_members
 - selectedProject.php, 34
- get_all_project_members_and_master
 - sprints.php, 39
- get_all_project_without_user_id
 - projects.php, 25
- get_all_sprints
 - sprints.php, 40
- get_all_task_inside_sprint
 - sprints.php, 40
- get_all_todo_doc
 - doc.php, 12
- get_all_us_inside_project
 - sprints.php, 41
- get_all_us_inside_sprint
 - sprints.php, 41
- get_all_us_inside_task
 - sprints.php, 41
- get_all_user_not_in_project
 - utility.php, 53
- get_git_url
 - release.php, 29
- get_last_commit
 - release.php, 29
- get_project_by_id
 - selectedProject.php, 34
- get_project_master
 - selectedProject.php, 35
- getAllUsers
 - modPanel.php, 17
- getCategories
 - faq.php, 14
- getCategoryId
 - faq.php, 15
- getQA
 - faq.php, 15
- getUserInvitationsOrRequest
 - profile.php, 21
- getUserNbParticipation
 - profile.php, 21
- getUserProfile
 - profile.php, 22
- is_master
 - selectedProject.php, 35
- is_member
 - selectedProject.php, 36
- leave_a_project
 - projects.php, 26
- link_us_to_sprint
 - sprints.php, 42

link_us_to_task
sprints.php, 42

modPanel.php
changeUserRole, 17
deleteUser, 17
getAllUsers, 17

modify_inside_project_US
backlog.php, 6

modify_inside_project_role
backlog.php, 5

pass_all_tests
tests.php, 51

profile.php
AcceptInvitation, 19
CancelRequest, 20
deleteAccount, 20
editInfo, 21
getUserInvitationsOrRequest, 21
getUserNbParticipation, 21
getUserProfile, 22

projects.php
add_invitation_request, 23
associate_project_and_user, 23
check_if_request_already_exist, 24
create_new_project, 24
get_all_project_by_user_id, 25
get_all_project_without_user_id, 25
leave_a_project, 26
remove_by_project_id, 26

release.php
change_git_url, 27
delete_all_commits, 28
deprecate_all_test, 28
get_all_commit, 28
get_git_url, 29
get_last_commit, 29
save_all_commit, 30

remove_US_by_id
backlog.php, 7

remove_by_project_id
projects.php, 26

remove_by_role_id
backlog.php, 6

remove_task
sprints.php, 43

save_all_commit
release.php, 30

searchQA
faq.php, 15

selectedProject.php
acceptRequest, 31
deleteInvitationOrRequest, 31
deleteMember, 33
get_all_project_invitations, 33
get_all_project_joining_requests, 34
get_all_project_members, 34
get_project_by_id, 34
get_project_master, 35
is_master, 35
is_member, 36

sprints.php
add_sprint_background, 37
count_nb_task_by_state_in_sprint, 38
create_new_sprint, 38
create_new_task, 38
delete_sprint_by_id, 39
get_all_project_members_and_master, 39
get_all_sprints, 40
get_all_task_inside_sprint, 40
get_all_us_inside_project, 41
get_all_us_inside_sprint, 41
get_all_us_inside_task, 41
link_us_to_sprint, 42
link_us_to_task, 42
remove_task, 43
switch_task_state, 43
unlink_us_from_sprint, 44
unlink_us_from_task, 44
update_task, 44

switch_task_state
sprints.php, 43

tests.php
add_new_test, 46
change_state, 46
check_deprecated, 47
compute_proportion, 47
count_proportion, 48
delete_test, 48
edit_test, 49
get_all_deprecated_tests, 49
get_all_failed_tests, 50
get_all_never_run_tests, 50
get_all_passed_tests, 50
pass_all_tests, 51

unlink_us_from_sprint
sprints.php, 44

unlink_us_from_task
sprints.php, 44

update_task
sprints.php, 44

utility.php
add_invitation_by_user_name, 52
check_if_request_already_exist, 52
get_all_user_not_in_project, 53