

# forumControl

Created by: peterdownie.com

Software Version: 2.41g

Greater Code Name: Atomic

Lesser Code Name: Proton

Service Version: 1.01

## Purpose

forumControl is a forum system. It does not include users, userControl was made for that and in the "

future there will be tripCodeControl.

Very happy to release this as BETA 1.01!

March 15 2017 Update

I will be updating this to have sub forums or threads or such.

This will have a discussion about other web services in the future

as well as allowing talking about chess which was its original purpose.

## lowerText

## technicalSpecifications

Possible status values  
ERROR: something has gone wrong. Returns severity, which is how dangerous the error is.  
NOMINAL: no errors were detected  
severity: Only returned when status is ERROR  
MINOR: Happens with normal use  
MODERATE: Something happened which should not.  
MAJOR: There is a serious problem.

## Main Methods

\_\_construct

createForum

deleteAForum

commentOnForum

deleteAcommentOnAForum

hideAComment

getPost

getAllPostOnForum  
getSomePostsOnForum  
getSomePostsOnForum\_ActivePostsOnly  
unHideComment  
getForumOwnerData  
getAllForumsOwnedByUser  
getAllForumsCommentedOnByUser  
get\_X\_MostRecentForums  
updateUsername  
getForumsBelowMax  
getForumsAboveMin  
getCurrentMaxForumValueInProject  
searchForums

## Method Details

### **\_\_construct**

*Parameter #0 [ <required> \$host ]*

*Parameter #1 [ <required> \$user ]*

*Parameter #2 [ <required> \$password ]*

*Parameter #3 [ <required> \$database ]*

*Parameter #4 [ <required> \$prePend ]*

*Parameter #5 [ <required> \$projectIdentification ]*

Construct a new object

### **createForum**

*Parameter #0 [ <required> \$userID ]*

*Parameter #1 [ <required> \$username ]*

*Parameter #2 [ <required> \$forumName ]*

*Parameter #3 [ <required> \$forumIntroText ]*

allows the creation of a forum, you will need to make sure that you already have a primary keyfunction, or database with that value.

This is split to reduce complexity and ease intergration with other systems. ConnectionID is for making sure that the database is paired properly

### **deleteAForum**

*Parameter #0 [ <required> \$forumID ]*

deletes a forum but makes sure that checks are made before deleting the data. You can use the `deleteAForum_NonSAFE` if you wish to delete without these checks.

## **commentOnForum**

*Parameter #0 [ <required> \$userID ]*

*Parameter #1 [ <required> \$forumID ]*

*Parameter #2 [ <required> \$comment ]*

Add a comment to a forum.

## **deleteAcommentOnAForum**

*Parameter #0 [ <required> \$postID ]*

This version Allow you to have comments deleted. Deleting comments may cause problems.

## **hideAComment**

*Parameter #0 [ <required> \$postID ]*

Hides a comment from view in a forum. This is a better idea than deleting posts as deleting them may cause problems.

## **getPost**

*Parameter #0 [ <required> \$postID ]*

Get a post within a specific project.

## **getAllPostOnForum**

*Parameter #0 [ <required> \$forumID ]*

This will get all of the posts on a forum. You can also specify which posts you want

to get in the function called `getSomePostsOnForum()`. This function is not controlled by ['output']

Instead it will return all rows for a forumID. Each post will be pushed onto an output array

## **getSomePostsOnForum**

*Parameter #0 [ <required> \$forumID ]*

*Parameter #1 [ <required> \$entryMin ]*

*Parameter #2 [ <required> \$entryMax ]*

This will allow you to get some posts from a forum. It returns all data in the database"

for the forum between the entries. It does use output as a controlling array type.

### **getSomePostsOnForum\_ActivePostsOnly**

*Parameter #0 [ <required> \$forumID ]*

*Parameter #1 [ <required> \$startEntry ]*

*Parameter #2 [ <required> \$MaxCount ]*

gets an array of active post, you can count these and if the count is maxCount there

are likely more results, if there are less than there likely are no results or no results at this time.

### **unHideComment**

*Parameter #0 [ <required> \$postID ]*

Changes a comment status from DISABLED to ACTIVE. Makes getSomePostsOnForum\_ActivePostsOnly see the comment again.

### **getForumOwnerData**

*Parameter #0 [ <required> \$forumID ]*

Get the user\_ID to connect to a remote database and the username of that user. The username is only the last used and may not be the current username.

### **getAllForumsOwnedByUser**

*Parameter #0 [ <required> \$userID ]*

Gets all of the forumIDs created by a user ID. UserID should be linked to an external control.

### **getAllForumsCommentedOnByUser**

*Parameter #0 [ <required> \$userID ]*

gets All of the forums that a user has commented on.

### **get\_X\_MostRecentForums**

*Parameter #0 [ <required> \$count ]*

gets the most recently created Forums.

## **updateUsername**

*Parameter #0 [ <required> \$userID ]*

*Parameter #1 [ <required> \$new\_username ]*

Change the stored username of a forum. The username is for temp use, not for exact data.

## **getForumsBelowMax**

*Parameter #0 [ <required> \$maximum\_exclusive ]*

*Parameter #1 [ <required> \$count ]*

Gets count Items below the maximum exclusive values(Exclusive means it does not include maximum).

## **getForumsAboveMin**

*Parameter #0 [ <required> \$minimum\_exclusive ]*

*Parameter #1 [ <required> \$count ]*

This gets X amount of forums about minimum exclusive, you can feed the returned number back into this function recursively. If you want to start at the first one use 0.

## **getCurrentMaxForumValueInProject**

This will get the amx value of the forums for use in the function getForumsBelowMax

## **searchForums**

*Parameter #0 [ <required> \$searchQuery ]*

Searchs through all of the forums name, introduction and psot data. This uses regular expressions which is both powerful and more complicated I would like. This

returns up to 4 queries. See tech for more details.

# **Method Technical Specifications**

# \_\_construct

Required Parameters  
host: The url where the database is hosted.

user: The user for the database.

password: The password for the user of the database.

database: The database name on the host url.

prePend: prePend is used for storing multiple tables that may conflict in one database.

projectIdentification: Project ID is for storing multiple projects in one table. This reduces performance but may be worth it if you want to use cheap servers.

Optional Values  
as cookiesport: The port of the database

socket: The socket of the database

## createForum

Usage: UserID, username and project\_identification ID all come from the user control table(or other system with the same idea). The rest does not need explaining. This function will generate a bunch of different status(an possibly severity, see class technical specifications) and output. red outputs are when status is ERROR, green outputs are returned on status of NOMINAL

output

INTRO\_OVER\_CHAR\_LIMIT(MINOR) when this is triggered it will also return a ['introLimit'] of 1500

FORUM\_NEEDS\_A\_NAME(MINOR) You can not create a forum without a name.

USER\_ALREADY\_HAS\_A\_FORUM\_WITH\_SAME\_NAME(MINOR) will be triggered if the userID or username already have a forum with the same forum name

FAILED\_CREATION(MINOR) no rows were inserted. An ['output'] of DATABASE\_ERROR means that more than one rows were returned, this should not be possible.

CREATED\_SUCCESSFULL means that the forum was created. With this it will also return the ['forumName']

DATABASE\_ERROR:(MAJOR) Too many rows inserted. This should never happen.

## deleteAForum

Usage

Returns output as well as status(and possibly severity). red outputs are when status is ERROR, green outputs are returned on status of NOMINAL

Output values

CANNOT\_DELETE\_NON\_EXISTENT\_FORUM(MINOR) when no rows are found.

DATABASE\_CORRUPTED\_SHORT(MAJOR) when more than one row is returned, this should never happen.

DELETION\_FAILED(MINOR) than it had the right parameters but the deletion failed, this could be because of improper credentials.

FORUM\_DELETED and if this happens it will return a forumID of the forum that was deleted.

TOO\_MANY\_DELETED(MAJOR): Too many forums were deleted.

# commentOnForum

## Usage

As with most of my functions the main output is ['output'], status and possibly severity. and depending on the outputed value there can be other array elements returned. red outputs are when status is ERROR, green outputs are returned on status of NOMINAL

## Possible Output Values

FORUM\_ID\_DOES\_NOT\_EXIST\_IN\_PROJECT(MINOR): Forum ID does not exist in the project,

FORUM\_ID\_TOO\_MANY\_IN\_PROJECT(MAJOR): Project has too many forum IDs in the project.

TOO\_MANY\_CHARACTERS(MINOR) than the person who installed the database decided that the max amount of characters is X. It will return ['maxCharacters'] as well.

NOT\_ENOUGH\_CHARACTER(MINOR): than it will return ['minCharacters'] as well.

COMMENT\_FAILED\_INSERT(MINOR): Than there is likely an error with the control or the database.

COMMENT\_SUCCESSFULLY\_INSERTED: than the comment was successfully inserted. Other data is returned, see below

Data returned when output is COMMENT\_SUCCESSFULLY\_INSERTED and/or status is NOMINAL  
forumID: The forum  
Posted to

post: The data posted

Post\_ID: The post ID

projectIdentificationID: The Project Identification.

# deleteAcommentOnAForum

## Usage

['output'] is the controlling output in this function and depending on its value there can be other array elements. This function also has status and on status ERROR has severity. red outputs are when status is ERROR, green outputs are returned on status of NOMINAL

## Possible Output Values

POST\_NUMBER\_DOES\_NOT\_EXIST(MINOR) the item you are trying to delete has already been deleted.

DATABASE\_ERROR\_MULTIPLE\_POST\_NUMBER\_IN\_PROJECT(MAJOR): multiple rows have been returned. This is a serious error and likely means the database has been corrupted.

POST\_DOES\_NOT\_MATCH\_PROJECT(MINOR): You cannot delete a post in a different project.

POST\_HAS\_TOO\_MANY\_PROJECTS(MAJOR): Post has too many projects, this should never happen.

FAILED\_TO\_DELETE\_ENTRY(MINOR) means that the database is likely not configured right.

DELETED\_MULTIPLE\_ROWS\_DATABASE\_IS\_CORRUPT(MAJOR) is when the database deletes multiple rows this should never happen and if it does it means the the database is corrupt.

DELETED\_SUCCESSFULLY and when ['output'] has this value it will also output a ['postID'] of what has been deleted

# hideAComment

Usage Again this is controlled by ["output"] and depending on the value than other arrays may be returned. This function also has status and on status ERROR has severity. red outputs are when status is ERROR green outputs are returned on status of NOMINAL

Possible Output values

POST\_ID\_DOES\_NOT\_EXIST(MINOR) means the post does not exist.

MULTIPLE\_KEYS\_RETURNED\_DATABASE\_CORRUPT(MAJOR) means that too many posts were returned. This should never happen and indicates a problems with the code or a corrupt database.

POST\_NOT\_IN\_PROJECT(MINOR): The post is in a different project. You cannot modify other projects.

TOO\_MANY\_POSTS\_IN\_PROJECT(MAJOR) Too many posts in one project. This indicates a database error. Likely it was not setup properly.

ALREADY\_DISABLED means no change is needed and you may want to code this as a success.

FAILED\_TO\_DELETE(MINOR) means that there is likely a database error or database permission error.

UPDATED\_TOO\_MANY\_ROWS\_DATABASE\_CORRUPT(MAJOR) means there is an error in the code or more likely a problem with the database.

SUCCESSFULLY\_CHANGED\_TO\_DISABLED means that it was successful and when it is this value it also returns ["postID"] with the disabled post

# getPost

The return is controlled by an ['output'] and status, with the possibility of severity if status is ERROR(Red outputs are on ERROR), green outputs are on NOMINAL status.

Possible output values

ID\_NOT\_FOUND(MINOR): ID not found in database

TOO\_MANY\_ID\_FOUND\_DATABASE\_CORRUPTION(MAJOR): Too many IDs found in database. This indicates that the database has or was not setup properly.

POST\_NOT\_IN\_PROJECT(MINOR): The post was found but within a different project. You want items in your projects.

TOO\_MANY\_POSTS\_IN\_PROJECT(MAJOR): Too many IDs with projects found. This should never happen.

POST\_FOUND\_SUCCESSFULLY: The post was found. It will return post and status

When Output is POST\_FOUND\_SUCCESSFULLYpost: What was posted

Poststatus: When this was written there were two states: active and disabled. Disabled is a better way then deleting an item.

creatorID: The user that created the post. Stored externally.

# getAllPostOnForum



This function outputs an output variable and status(and severity if status is ERROR), Green items are NOMINAL status and Red items are ERROR status.

Possible Output Data

FORUM\_DOES\_NOT\_EXIST(MINOR): Common error when something was not added to the database.

TOO\_MANY\_FORUMS\_EXIST(MAJOR): Indicates the database was not setup properly or has been corrupted.

FORUM\_EXISTS\_BUT\_NOT\_IN\_PROJECT(MINOR): The forum you are trying to access is in a different project and should not be accessed.

FORUM\_IN\_TOO\_MANY\_PROJECTS(MAJOR): This Indicates that the database has been setup improperly or has been corrupted.

THERE\_ARE\_NO\_POSTS: Forum was found but there is nothing to output.

OUTPUTTING\_FEED: Posts found, will be outputting data. When output is OUTPUTTING\_FEED

If table has been modified and this has not been updated than. Will return multiple rows. post\_ID: Identification to retrieve.

forum\_fk The fk forumID

user\_fk: The user that created the value, references remote databases.

entryNumber: What entry this is. Starts at one. Some may be skipped if deleted or hidden.

entry\_date: The day the comment was made

post: The comment itself.

status: Whether the item is active or not.

# getSomePostsOnForum

This function is controlled by output, if output is not outputted than it will start feeding out rows of data. Red Values are where status is ERROR

Possible output values

FORUM\_DOES\_NOT\_EXIST(MINOR): Forum does not exist

TOO\_MANY\_FORUMS\_EXIST(MAJOR): Too many forums exist. This indicates the database has been setup incorrectly.

FORUM\_EXISTS\_BUT\_NOT\_IN\_PROJECT(MINOR): The forum was found but in a different project. You should only access items in your project.

FORUM\_IN\_TOO\_MANY\_PROJECTS(MAJOR): The forum was found in multiple projects. This indicates the database was setup incorrectly.

NO\_ITEMS\_IN\_RANGE: This indicates that the item was found successfully but there was no data.

OUTPUTTING\_FEED: Will output more data in feed

When output is OUTPUTTING\_FEED

If table has been modified and this has not been updated than. Will return multiple rows. post\_ID: Identification to retrieve.

forum\_fk The fk forumID

user\_fk: The user that created the value, references remote databases.

entryNumber: What entry this is. Starts at one. Some may be skipped if deleted or hidden.

entry\_date: The day the comment was made

post: The comment itself.

status: Whether the item is active or not.

# getSomePostsOnForum\_ActivePostsOnly

This gets the active posts. It is controlled by a array element called output. If output is not set than it dumps out the post data. It returns output and status. It also returns severity if status is ERROR. RED outputs are on erroand green outputs are on NOMINAL

Possible output values

FORUM\_DOES\_NOT\_EXIST: Forum does not exist in the database.

TOO\_MANY\_FORUMS\_EXIST: Too many forums were found, this is quite a bit more serious than the FORM\_DOES\_NOT\_EXIST

FORUM\_EXISTS\_BUT\_NOT\_IN\_PROJECT: Forum was found in a different project.

FORUM\_IN\_TOO\_MANY\_PROJECTS: Multiple forums were found in one project. This is quite serious.

NO\_ROWS\_RETURNED: No rows were returned.

OUTPUTTING\_FEED: Rows found, will output

If output is OUTPUTTING\_FEED will return one or more rows.post\_ID: The id used to store in the database.

forum\_fk: The froum that it is part of. This should always be teh inputted forum Id.

user\_fk: The user ID that posted the data. This is linked to an external soap function.

entryNumber: The entry number. It is incremented by one when posting. This will not always be in order becasue some items are hidden.

entry\_date: The date the entry was made.

post: The entry itself.

status: The status of the post. This is different than the database status.

# unHideComment

Unhide a comment changes the comments status from DISABLED to ACTIVE. It returns an output variable that tells you what has happend.

Possible Output Values

POST\_ID\_DOES\_NOT\_EXIST(MINOR): Post was not found in the database.

MULTIPLE\_KEYS\_RETURNED\_DATABASE\_CORRUPT(MAJOR): Post was found multiple times in the database. This means that the database was likely not setup properly.

POST\_COULD\_NOT\_FIND\_FORUM\_ID(MODERATE): This indicates that the post was found but the forumID was not setup wrong. Not really sure why you would ever get this error.

TOO\_MANY\_POST\_AND\_FORUM\_FK(MAJOR): This indicates that there were multiple posts found multiple times. It should really have come up with MULTIPLE\_KEYS\_RETURNED\_DATABASE\_CORRUPT. This likely indicates that the SPROC was modified improperly.

FORUM\_DOES\_NOT\_EXIST\_IN\_DATABASE(MODERATE): The post was found but the forum it was created under

does not exist. It most likely has been deleted but it is also possible that there is an error with the code.

TOO\_MANY\_FORUM\_EXISTS(MAJOR): Multiple forums of the same were found. This indicates that the database was setup improperly as it should not allow multiple primary keys.

FORUM\_EXISTS\_BUT\_NOT\_IN\_PROJECT(MINOR): The forum was found but it is in a different project. You should only access your part of the project. This was added to allow multiple forum controls to use the same table names.

FORUM\_EXISTS\_BUT\_TOO\_MANY(MAJOR): The forum was found in the project but too many entries. I am not sure why you would ever get this error but it indicates a major problem.

ALREADY\_ACTIVE: There is no need to unhide the comment as it is already unhid.

FAILED\_TO\_CHANGE(MINOR): The database did not allow the item to be changed. This is likely a permission error.

UPDATED\_TOO\_MANY\_ROWS\_DATABASE\_CORRUPT: There were multiple rows updated. This indicates the database was not setup properly.

SUCCESSFULLY\_CHANGED\_TO\_ACTIVE: Successfully changed the item from DISABLED to ACTIVE.

## getForumOwnerData

getForumOwnerData is controlled by a array element called output. If output is not set than it is retrieving data.

Possible output values

FORUM\_ID\_NOT\_FOUND(MINOR): Forum does not exist in the database.

TOO\_MANY\_FORUMS\_FOUND(MAJOR): Too many forum found. This indicates that the database is corrupted.

FORUM\_EXISTS\_NOT\_IN\_PROJECT(MINOR): The forum was found but it is not in the project. You should only access things in your project. This allows multiple projects to be contained in the same database tables.

FORUM\_EXISTS\_IN\_PROJECT\_MULTIPLE\_TIMES(MAJOR): The project was found in too many projects. I am not sure why it did not come up with TOO\_MANY\_FORUMS\_FOUND but this indicates the project is corrupted or not installed properly.

EVERYTHING\_OKAY: Row was found, will output data.

If output is EVERYTHING\_OKAY than these are also returned.creatorID: The user ID who made the forum

username: The username who created the forum. This is for reference only. It should not be relied upon,

## getAllForumsOwnedByUser

This function is controlled by an output variable. Green items are when status is NOMINAL and Red items are when status is ERROR

Possible output values

NO\_ENTRIES\_IN\_THE\_DATABASE\_FOR\_CREATOR\_ID: The database has no entries for that

userID/>ENTRIES\_FOUND\_NOT\_IN\_PROJECT: There were entries in the database with that creator. However they were in a different project. You should only use items in your project.

OUTPUTTING\_FEED: Will return feed of below.

When output is OUTPUTTING\_FEED than will return feed of below.forum\_ID: ID of the forum in the database.

projectIdentificationID: The project that it is in.

name: Name of the forum

creatorID: The ID of the creator.

username: The creator's username. For reference, may be different in the referenced database.

introduction: What the user wants the forum to be about.

creationDATE: When the forum was created

lastEntryDATE: The last time an entry was added.

status: Can be either ACTIVE or DISABLED, DISABLED is not shown in active only.

# getAllForumsCommentedOnByUser

If there is an error, output is returned. Otherwise forums in the project are returned.

Possible output values

NO\_FORUM\_COMMENTED\_BY\_USER\_ID(MINOR): There are no entries for given userID.

NO\_FORUM\_COMMENTED\_BY\_USER\_ID\_IN\_PROJECT(MINOR): There are entries for given user but none are in this project. You should stay in your own project.

NO\_COMMENTS\_ON\_FORUMS(MINOR): There were no comments on the forum.

OUTPUTTING\_FEED: Will be outputting all of the data successfully.

Other data will come in feed[var] when status[output] is OUTPUTTING\_FEEDtempID: Temp ID for linking number, you can ignore this.

forum\_ID: The forum\_ID of the database element.

name: The name of the forum.

introduction: An introduction to the forum.

# get\_X\_MostRecentForums

If output is set it is NO\_ROWS\_RETURNED and no items will be returned. Otherwise returns an array of most recent forums created by date.

NO\_ROWS\_RETURNED: No rows were found in the database.

OUTPUTTING\_FEED: Will return items listed below.

Data will be output in feed when [output] variable is OUTPUTTING\_FEEDforum\_ID: ID of the forum in the database.

projectIdentificationID: The project that it is in.

name: Name of the forum

creatorID: The ID of the creator.

username: The creator's username. For reference, may be different in the referenced database.

introduction: What the user wants the forum to be about.

creationDATE: When the forum was created

lastEntryDATE: The last time an entry was added.

status: Can be either ACTIVE or DISABLED, DISABLED is not shown in active only.

# updateUsername

updateUsername is controlled by a array element called output. GREEN items are when status is NOMINAL and RED items are when the status is ERROR

output values

NO\_USER\_ID\_IN\_DATABASE: User Id was not found in the database.

NO\_USER\_ID\_IN\_PROJECT: User ID was found in the database but not in this project. You should only change items in your project.

NO\_USER\_NEEDS\_UPDATING: The data is already the new\_username you have tried to set.

NO\_ROWS\_UPDATED: The item needed updating but it did not update. This is likely a minor permission problem.

UPDATED\_SUCCESSFULLY: Items were updated.

# getForumsBelowMax

This function is controlled by a variable called status.

Possible output values

NO\_ITEMS\_IN\_DATABASE: The database(table) is not populated.

NO\_ITEMS\_IN\_PROJECT: There are items in the database but none about your project.

NO\_ITEMS\_UNDER\_MAX: There are in the database in your project but your maximum is too low to retrieve any of them.

COUNT\_IS\_ZERO: You will not load any data because a limit of 0 will always return no results.

NO\_RESULTS: There were no results under your specified value.

OUTPUTTING\_FEED: The next query will return the structure listed above.

When output is OUTPUTTING\_FEED lastResult: Returns the last result. You can use this to feed the maximum.

moreResults: If there are more items in the database. Is either MORE\_RESULTS or END\_OF\_THE\_LINE

When Nominal it will feed out rows of data, of the following structure in the next query. forum\_ID: ID in database.

projectIdentificationID: The project the forum is in.

name: Name of the forum

creatorID: Who created the forum.

username: The username who made the forum. This is for reference only and should not be trusted..

introduction: An introduction on why the forum was built.

creationDATE: When the forum was created.

lastEntryDATE: The last time someone added an entry.

status: The status of the database. Either ACTIVE or DEACTIVATED.

Every return has an output which gives you additional information about what is going on in the database. RED items are returned WHEN status is ERROR. GreenItems are returned when status is NOMINAL

# getForumsAboveMin

This function is modeled after the function `getForumsBelowMax`. This function is controlled by a variable called `status`.

Possible output values

`NO_ITEMS_IN_DATABASE`: The database(table) is not populated.

`NO_ITEMS_IN_PROJECT`: There are items in the database but none about your project.

`NO_ITEMS_OVER_MIN`: There are no items in the project over the min amount.

`COUNT_IS_ZERO`: You will not load any data because a limit of 0 will always return no results.

`NO_RESULTS`: There were no results under your specified value.

`OUTPUTTING_FEED`: The next query will return the structure listed above.

When `status` is output is `OUTPUTTING_FEED` `lastResult`: Returns the last result. You can use this to feed the maximum.

`moreResults`: If there are more items in the database. Is either `MORE_RESULTS` or `END_OF_THE_LINE`

When Nominal it will feed out rows of data, of the following structure in the next query.  
`forum_ID`: ID in database.

`projectIdentificationID`: The project the forum is in.

`name`: Name of the forum

`creatorID`: Who created the forum.

`username`: The username who made the forum. This is for reference only and should not be trusted..

`introduction`: An introduction on why the forum was built.

`creationDATE`: When the forum was created.

`lastEntryDATE`: The last time someone added an entry.

`status`: The status of the database. Either `ACTIVE` or `DEACTIVATED`.

Every return has an output which gives you additional information about what is going on in the database. RED items are returned WHEN `status` is `ERROR`. Green items are returned when `status` is `NOMINAL`

# getCurrentMaxForumValueInProject

No technical specifications found

## searchForums

`search forms` is controlled by a `status` variable. If Error there is only one query. If Nominal there is 4. When `status` is

`ERROR` Possible outputs

GREEN when `status` is `NOMINAL` RED when `STATUS` is `ERROR`

`NO_FORUMS_CREATED`: There are no items in the database forum table.

`NO_ITEMS_IN_PROJECT`: There are items in the database forum table but none are in your project. You should stay within your project.

`EVERYTHING_OKAY`: There are no obvious errors, operation will continue normally.

Forum Data: When the `status` is `NOMINAL`, the 2nd and 3rd returned queries will be this with 0 or more rows.  
`forum_ID`: ID in database.

`projectIdentificationID`: The project the forum is in.

`name`: Name of the forum

`creatorID`: Who created the forum.

username: The username who made the forum. This is for reference only and should not be trusted..

introduction: An introduction on why the forum was built.

creationDATE: When the forum was created.

lastEntryDATE: The last time someone added an entry.

status: The status of the database. Either ACTIVE or DEACTIVATED.

Forum Post: When the status is NOMINAL, the 4th return is 0 or more rows