

# userControl

Created by: peterdownie.ails to auto generate, working on a fixcom

Software Version: 2.41k

Greater Code Name: Atomic

Lesser Code Name: Neutron

Service Version: 1.02

## Purpose

userControl takes control of usernames to reduce complexity in chessToast

However it may be used with other programs in the future. You create a user and then you can

access these users and receive a primary key. This allows you to connect to remote databases

Technical Specifications Need Updating.

## lowerText

Created by Peter Downie of peterdownie.com

## technicalSpecifications

Optional Cookies: Not required but will allow additional functionality  
port: the port of the database if it is not the default.  
socket: the socket of the database if it is not the default.  
projectIdentification: HIGHLY RECOMMENDED with random and unique key for making connections easier. This replaces the user Connection key which is too cumbersome to use.  
prePend: the port of the database if it is not the default.

## Main Methods

\_\_construct

database\_setup

createNewUser

login

get\_username\_data

change\_user\_data

createRecoveryCode

verifyUsername  
recoverUsingTempCode  
getDataByUserID

## Method Details

### **\_\_construct**

This used to connect to the database. I think it is much better to have a function do this.

### **database\_setup**

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

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

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

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

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

Needs to be called for this web service to work.

Optional \$options can contain lowercase port, socket, projectIdentification and prePend

### **createNewUser**

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

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

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

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

Create new user allows you to create a user for the database and project combo

### **login**

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

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

Valid Login makes sure that a user with that password is in the database/project combination.

### **get\_username\_data**

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

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

Replaces a bunch of functions to make the webservice easier to learn

Request Values *<i>password\_hint, exists, validated</i>*

**<b>password\_hint:</b>** Retrieves the password hint for a given user.

Returns string of the password hint

**<b>exists:</b>** Checks to see if the given username is already in the database, returns a string

**<b>validated:</b>** returns a string which has 3 possible values.

## **change\_user\_data**

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

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

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

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

Allows changing of a users data!

changeType(s): email, password

## **createRecoveryCode**

*Parameter #0 [ <required> \$input\_array ]*

Create a recovery password, keys of *<i>username, or email</i>*

## **verifyUsername**

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

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

Verify makes sure that the user has the email address they stated.

## **recoverUsingTempCode**

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

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

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

Recovery using a temp code is a way of recovering a password when you have the email

address associated with the temp Code.

## **getDataByUserID**

*Parameter #0 [ <required> \$data\_requested ]*

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

possible data requests are *<i>username, email, lastSignInData and signUpData</i>*

## Method Technical Specifications

### \_\_construct

No technical specifications found

### database\_setup

No technical specifications found

### createNewUser

Returns a string with possible values listed below. Dark Red Items return extraInfo about the error

NAME\_IS\_RESERVED(MINOR): Name cannot be NOT\_FOUND.

ILLEGAL\_CHARACTER\_IN\_USERNAME(MINOR): Username cannot contain an @ symbol

USER\_ALREADY\_EXISTS(MINOR): Username already in database and project. You can have multiple users of the same as long as they are in different projects

EMAIL\_ALREADY\_IN\_DATABASE(MINOR): email already in database and project. Can have multiple emails as long as project name is different.

PASSWORD\_IS\_TOO\_LONG(MINOR): Password is longer than 50 characters

PASSWORD\_LENGTH\_IS\_TOO\_SHORT(MINOR): Password needs to be at least 8 characters.

RANDOM\_KEY\_GENERATION\_ERROR: This is a compound error, uses ending part. Random key was not generated.

SALT\_GENERATION\_ERROR: This is a compound error, uses ending part. Salt for password was not generated.

MUST\_BE\_AT\_LEAST\_1\_IN\_LENGTH(MINOR): Ending of a compound error. Item must be over 1 Character.

MAX\_LENGTH\_255\_CHARACTERS(MINOR): Ending of a compound error. Item must be equal or less than 255 Character.

USER\_FAILED\_CREATION(MINOR): The user was not created. Likely a permission problem.

USER\_SUCCESFULLY\_CREATED: User was successfully created.

When output is in the dark red of RANDOM\_KEY\_GENERATION\_ERROR or SALT\_GENERATION\_ERROR than it returns an addition extraInfo of

Possible extraInfo data

MUST\_BE\_AT\_LEAST\_1\_IN\_LENGTH(MINOR): password is too short

MAX\_LENGTH\_255\_CHARACTERS:(MINOR): password is too long.

# login

Always returns output, depending on output may return other items as well.

Possible ['output'] values

USER\_PROJECT\_COMBO\_NOT\_FOUND\_IN\_DB: username with project was not found in the database.

USER\_PROJECT\_COMBO\_FOUND\_WRONG\_PASSWORD: username with project was found but your password was wrong.

TOO\_MANY\_USER\_PROJECT\_COMBO\_FOUND: too many username/project was found without password, there is problems with the database.

TOO\_MANY\_USERS\_FOUND\_DB\_ERROR: too many username/project was found without password, there is problems with the database.

QUERY\_FAILED: There was an error with the query

VALID\_LOGIN: User was found in the database. Also returns other items listed below.

When ['output'] is VALID\_LOGINidKey: Used for connecting to other items.

# get\_username\_data

Request: password\_hint Possible output values

USERNAME\_NOT\_FOUND(MINOR): User was not found in the database.

USERNAME\_FOUND\_BUT\_NOT\_IN\_PROJECT(MINOR): username was found in the database but not in the same project. This may indicate that you have a projectID error

TOO\_MANY\_USERNAMES\_DATABASE\_CORRUPT(MAJOR): Too many users were found in the project.

SUCCESS: Username was found. Additional output of passwordHint is returned.

Request: existsalways returns an ['output'] may return other depending on the value of the ['output'] string return

USERNAME\_NOT\_FOUND(MINOR): Username is not in the database.

USERNAME\_FOUND\_BUT\_NOT\_IN\_PROJECT(MINOR): Username was found but not in the current project. This

means you may have misspelled your projectID.

USERNAME\_FOUND\_IN\_DATABASE: Username is in the database.

TOO\_MANY\_USERS\_FOUND\_IN\_DATABASE(MAJOR): Too many usernames found in database.

Request: password\_hint Returns USERNAME\_NOT\_IN\_DATABASE(MINOR): The username is not in the database.

USERNAME\_NOT\_IN\_PROJECT(MINOR): The username exists but is not in the current project.

ALREADY\_VALIDATED: Username is validated.

NEED\_TO\_VALIDATE: Username needs to be validated.

# change\_user\_data

email

newDataArray must have a key of email and a value of a valid email

This function allows you to change your email address, you will need your username and unsalted Password.

password

newDataArray must have a key of password, password\_hint

Change Password allows you to change the password of a user, it requires that you know the old password

It is also important to know that to change the password, it must be unsalted

changeEmailAddress always returns a string

Possible string outputs

USERNAME\_NOT\_IN\_DATABASE(MINOR): Username is not in the database.

USERNAME\_IN\_DIFFERNT\_PROJECT(MINOR): Username is in the database but not in your project. You might be in the wrong project.

TOO\_MANY\_ROWS\_WITH\_USERNAME(MAJOR): Username was found too many times in project. There are database issues.

USERNAME\_PASSWORD\_COMBO\_INVALID(MINOR): Username and password combo was not found in project.

TOO\_MANY\_USERNAME\_PASSWORD\_COMBO(MAJOR): Too many username/password found in project.

EMAIL\_IS\_TOO\_LONG\_MAX\_50\_CHARACTERS(MINOR): Your email is too long.

EMAIL\_ADDRESS\_MUST\_HAVE\_@\_SYMBOL(MINOR): Email does not include required @(at) symbol.

EMAIL\_AND\_NEW\_EMAIL\_MATCH\_NO\_CHANGE\_NEEDED(MINOR): email is already that value.

UPDATED\_EMAIL: Email successfully updated.

FAILED\_TO\_UPDATE\_EMAIL(MODERATE): Failed to update email. Likely a database permission problem.

Password: Always returns a string.

Possible string output

NO\_USER\_BY\_THAT\_NAME\_IN\_DATABASE(MINOR): The username was not found in the database.

NO\_USER\_BY\_THAT\_NAME\_IN\_PROJECT(MINOR): No username found in database for the particular project.

TOO\_MANY\_USERNAME\_IN\_PROJECT(MAJOR): More than one username found in the same project, there is a database error.

PASSWORD\_MUST\_BE\_LESS\_THAN\_50\_CHARACTERS(MINOR): Password cannot be longer than 50 characters.

PASSWORD\_MUST\_BE\_8\_OR\_MORE\_CHARACTERS(MINOR): Password must be at least 8 characters.

USERNAME\_AND\_PASSWORD\_DO\_NOT\_MATCH(MINOR): Username and old password does not match.

TOO\_MANY\_USERNAME\_PASSWORD\_PROJECT\_COUNT(MAJOR): Too many users with password found.

Database error.

NEW\_AND\_OLD\_PASSWORD\_ARE\_THE\_SAME(MINOR): Cannot set your password to the same value.

PASSWORD\_CHANGED: Password was successfully changed.

MORE\_THAN\_ONE\_ROW\_UPDATED\_DB\_ERROR(MAJOR): More than one username was changed. This means there is a database error.

PASSWORD\_CHANGE\_FAILED(MODERATE): Password failed to be changed. Likely a permission error.

# createRecoveryCode

Two merged functions need documenting

# verifyUsername

Verify username always returns a string

Possible string values

USERNAME\_NOT\_FOUND\_IN\_DATABASE(MINOR): username was not found in all of the database.

USERNAME\_NOT\_FOUND\_IN\_PROJECT(MINOR): username not in project/database combo. It is in the database, you may have connected to the wrong project.

TOO\_MANY\_USERNAMES\_FOUND\_IN\_PROJECT(MAJOR): Too many usernames found in project/database combo.

ALREADY\_VERIFIED: No need to verify an already verified account.

TOO\_MANY\_VERIFIED(MAJOR): Too many rows in the database have already been verified.

USERNAME\_VERIFICATION\_CODE\_NOT\_FOUND(MINOR): username was found in project but the verification code is different.

TOO\_MANY\_ROWS\_WITH\_VERIFICATION\_CODE\_FOUND(MAJOR): More than one row was found with username/verification code in project.

VERIFY\_SUCCESSFUL: Verified!

VERIFICATION\_UPDATE\_FAILED(MODERATE): Failed to verify, likely a database permission problem.

TOO\_MANY\_ROWS\_UPDATED(MAJOR): Too many rows were updated. Database problems.

# recoverUsingTempCode

recoverUsingTempCode always returns a string

Possible String outputs

USERNAME\_DOES\_NOT\_EXIST\_IN\_DATABASE(MINOR): The username is not in the database.

USERNAME\_DOES\_NOT\_EXIST\_IN\_PROJECT(MINOR): username not in database/project combo.

TOO\_MANY\_USERNAMES\_FOUND\_IN\_PROJECT(MAJOR): Too many usernames in the project. There is a database error.

TEMP\_CODE\_NOT\_CREATED(MINOR): Cannot recover as the temp code has not been set.

TOO\_MANY\_ROWS\_WITH\_TEMP\_CODE(MAJOR): There are too many temp codes. The database was not setup or is corrupt.

USERNAME\_VERIFICATION\_CODE\_DOES\_NOT\_MATCH(MINOR): You need the proper code to recover the password.

USERNAME\_VERIFICATION\_CODE\_TOO\_MANY\_MATCHES(MAJOR): Too many verification/user matches found in database/project. Database has issues.

SUCCESSFULLY\_CHANGED\_PASSWORD: Password was changed.

TOO\_MANY\_ROWS\_UPDATED(MAJOR): Too many rows updated. The database has issues.

FAILED\_TO\_CHANGE\_PASSWORD(MODERATE): Did not change password. Likely database permission error.

# getDataByUserID

4 FUNCTIONS NEED DOCUMENTING