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 recieve 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 functionalityport: 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>
password_hint: Retrieves the password hint for a given user. Returns string of the password hint
exists: Checks to see if the given username is already in the database, returns a string
validated: 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 mispelled your projectID.

USERNAME_FOUND_IN_DATABASE: Username is in the database.

TOO_MANY_USERS_FOUND_IN_DATABSE(MAJOR): Too many usernames found in database.

Request: password_hint ReturnsUSERNAME_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 outptus

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 alreayd 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