

My Project

Generated by Doxygen 1.8.6

Thu Apr 23 2015 12:47:33

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Namespace Documentation	7
4.1	userInfo Namespace Reference	7
4.1.1	Detailed Description	7
5	Class Documentation	9
5.1	test.TestStringMethods Class Reference	9
5.1.1	Detailed Description	9
5.1.2	Member Function Documentation	9
5.1.2.1	test_in_database	9
5.1.2.2	test_none_of_food_in_database_calculation	9
5.1.2.3	test_not_in_database	10
5.1.2.4	test_one_food_calculation	10
5.1.2.5	test_three_food_but_one_not_in_database_calculation	10
5.1.2.6	test_two_food_calculation	10
5.1.2.7	test_upper_lower_mix	10
5.2	userInfo.userInfo Class Reference	10
5.2.1	Detailed Description	10
5.2.2	Member Function Documentation	10
5.2.2.1	calculateCal	10
5.2.2.2	findFood	11
6	PHP Documentation	13
6.1	Documentation for PHP script	13
	Index	14

Chapter 1

Namespace Index

1.1 Namespace List

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

userInfo	7
------------------------------------	---

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

TestCase	
test.TestStringMethods	9
userInfo.userInfo	10

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

test.TestStringMethods	9
userInfo.userInfo	10

Chapter 4

Namespace Documentation

4.1 userInfo Namespace Reference

Classes

- class [userInfo](#)

4.1.1 Detailed Description

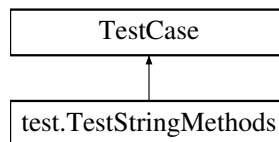
@package docstring

Chapter 5

Class Documentation

5.1 test.TestStringMethods Class Reference

Inheritance diagram for test.TestStringMethods:



Public Member Functions

- def `test_in_database`
- def `test_not_in_database`
- def `test_upper_lower_mix`
- def `test_one_food_calculation`
- def `test_two_food_calculation`
- def `test_three_food_but_one_not_in_database_calculation`
- def `test_none_of_food_in_database_calculation`

5.1.1 Detailed Description

This class uses to run the unit test for userInfo class

5.1.2 Member Function Documentation

5.1.2.1 def test.TestStringMethods.test_in_database (self)

This method uses to test if findFood function can find food in database

5.1.2.2 def test.TestStringMethods.test_none_of_food_in_database_calculation (self)

This method uses to test if calculateCal function can return correct calories for all food are not in database

5.1.2.3 `def test.TestStringMethods.test_not_in_database (self)`

This method uses to test if findFood function return false for food not in databases

5.1.2.4 `def test.TestStringMethods.test_one_food_calculation (self)`

This method uses to test if calculateCal function can return correct calories for one food

5.1.2.5 `def test.TestStringMethods.test_three_food_but_one_not_in_database_calculation (self)`

This method uses to test if calculateCal function can return correct calories for three food but not in database

5.1.2.6 `def test.TestStringMethods.test_two_food_calculation (self)`

This method uses to test if calculateCal function can return correct calories for two food

5.1.2.7 `def test.TestStringMethods.test_upper_lower_mix (self)`

This method uses to test if findFood function can find food that are mixed string (lower and upper characters)

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

- test.py

5.2 `userInfo.userInfo` Class Reference

Public Member Functions

- def [findFood](#)
- def [calculateCal](#)

Static Public Attributes

- dictionary **dictfood** = {'chickendrumstick':53.5,'egg':1.44,'milk':0.54}
- int **total** = 0

5.2.1 Detailed Description

This class uses to find food in our database or API and change users' information in databases
 @param dicfood It use to store information of food in case if we cannot use API
 @param total It use to store the total calories of user

5.2.2 Member Function Documentation

5.2.2.1 `def userInfo.userInfo.calculateCal (self, string)`

This method uses to check if the food name in our database or not.
 @param string It is the string name of food from the users.
 @return total calories

5.2.2.2 def userInfo.userInfo.findFood (*self*, *name*)

This method uses to check if the food name in our database or not.
@param name It is the name of the food from the users.
@return true if food in databases, false otherwise

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

- userInfo.py

Chapter 6

PHP Documentation

6.1 Documentation for PHP script

-login.php: received inputs(username,password,email) and protect it with stripslashes and real_escape_string to protect us from the code injection. submit query and if the query return only 1 line we start the session_register and print out login_success.

-login_html.php: a html file that has 3 input for login. and a search bar below.

-login_success.php: start session when success and print login success.

-logout.php: logout php

-signup.php: inserting into database, if user info is set we print successfully inserting.

-signup_html.php: a html page that has input boxes(username,email,security answer,security question,pet) and has action:post and connect with signup.php.

-user_main.html: simple search bar

-main.html: has a signup and login button with a search bar in the middle that will be connected to the python files.

Index

- calculateCal
 - userInfo::userInfo, [10](#)
- findFood
 - userInfo::userInfo, [10](#)
- test.TestStringMethods, [9](#)
- test::TestStringMethods
 - test_in_database, [9](#)
 - test_none_of_food_in_database_calculation, [9](#)
 - test_not_in_database, [9](#)
 - test_one_food_calculation, [10](#)
 - test_two_food_calculation, [10](#)
 - test_upper_lower_mix, [10](#)
- test_in_database
 - test::TestStringMethods, [9](#)
- test_none_of_food_in_database_calculation
 - test::TestStringMethods, [9](#)
- test_not_in_database
 - test::TestStringMethods, [9](#)
- test_one_food_calculation
 - test::TestStringMethods, [10](#)
- test_three_food_but_one_not_in_database_calculation
 - test::TestStringMethods, [10](#)
- test_two_food_calculation
 - test::TestStringMethods, [10](#)
- test_upper_lower_mix
 - test::TestStringMethods, [10](#)
- userInfo, [7](#)
- userInfo.userInfo, [10](#)
- userInfo::userInfo
 - calculateCal, [10](#)
 - findFood, [10](#)