

SocialPy Documentation

Generated by Doxygen 1.8.13

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	Authentication.Authentication Class Reference	5
3.1.1	Constructor & Destructor Documentation	5
3.1.1.1	__init__()	5
3.1.2	Member Function Documentation	6
3.1.2.1	login()	6
3.1.2.2	password_reset()	6
3.1.2.3	register()	6
3.2	CommandParser.CommandParser Class Reference	6
3.2.1	Detailed Description	7
3.2.2	Constructor & Destructor Documentation	7
3.2.2.1	__init__()	7
3.2.3	Member Function Documentation	8
3.2.3.1	parseCommand()	8
3.2.3.2	post_commands()	8
3.2.3.3	profile_commands()	8
3.3	firebase_creds.firebase_class Class Reference	9
3.3.1	Detailed Description	9
3.3.2	Constructor & Destructor Documentation	10
3.3.2.1	__init__()	10
3.3.3	Member Function Documentation	10
3.3.3.1	get_db()	10
3.3.3.2	get_UID()	10
3.3.3.3	get_user_instance()	11
3.3.3.4	get_username()	11
3.3.3.5	set_username()	11
3.3.3.6	sign_in()	11
3.3.3.7	sign_up()	12

4	File Documentation	13
4.1	src/Social Media App/CommandParser/CommandParser.py File Reference	13
4.1.1	Detailed Description	13
4.1.2	Variable Documentation	13
4.1.2.1	exit_text	14
4.2	src/Social Media App/CommandParser/help.py File Reference	14
4.2.1	Detailed Description	14
4.3	src/Social Media App/Firebase/Authentication.py File Reference	14
4.3.1	Detailed Description	15
4.4	src/Social Media App/Firebase/firebase_creds.py File Reference	15
4.4.1	Detailed Description	15
4.5	src/Social Media App/Posts/AddPost.py File Reference	15
4.5.1	Detailed Description	16
4.5.2	Function Documentation	16
4.5.2.1	addPost()	16
4.6	src/Social Media App/Posts/DeletePost.py File Reference	16
4.6.1	Detailed Description	17
4.6.2	Function Documentation	17
4.6.2.1	DeletePost()	17
4.7	src/Social Media App/Posts/EditPost.py File Reference	17
4.7.1	Detailed Description	18
4.7.2	Function Documentation	18
4.7.2.1	EditPost()	18
4.8	src/Social Media App/Posts/ViewPost.py File Reference	18
4.8.1	Detailed Description	19
4.8.2	Function Documentation	19
4.8.2.1	ViewPostsAll()	19
4.8.2.2	ViewPostsByUser()	19
4.8.2.3	ViewPostsFollowing()	20
4.9	src/Social Media App/Profile/AddFollowers.py File Reference	20

4.9.1	Detailed Description	20
4.9.2	Function Documentation	21
4.9.2.1	AddFollowers()	21
4.10	src/Social Media App/Profile/DeleteAccount.py File Reference	21
4.10.1	Detailed Description	21
4.10.2	Function Documentation	21
4.10.2.1	DeleteAccount()	22
4.11	src/Social Media App/Profile/DeleteFollowers.py File Reference	22
4.11.1	Detailed Description	22
4.11.2	Function Documentation	22
4.11.2.1	DeleteFollowers()	22
4.12	src/Social Media App/Profile/EditLocation.py File Reference	23
4.12.1	Detailed Description	23
4.12.2	Function Documentation	23
4.12.2.1	EditLocation()	23
4.13	src/Social Media App/Profile/EditName.py File Reference	24
4.13.1	Detailed Description	24
4.13.2	Function Documentation	24
4.13.2.1	EditName()	24
4.14	src/Social Media App/Profile/QueryFollowers.py File Reference	25
4.14.1	Detailed Description	25
4.14.2	Function Documentation	25
4.14.2.1	QueryFollowers()	25
4.14.2.2	QueryFollowing()	26
4.15	src/Social Media App/Profile/QueryProfile.py File Reference	26
4.15.1	Detailed Description	27
4.15.2	Function Documentation	27
4.15.2.1	QueryProfile()	27
4.16	src/Social Media App/Profile/ViewFollowers.py File Reference	27
4.16.1	Detailed Description	27
4.16.2	Function Documentation	28
4.16.2.1	ViewFollowers()	28
4.16.2.2	ViewFollowing()	28
4.17	src/Social Media App/Profile/ViewProfile.py File Reference	29
4.17.1	Detailed Description	29
4.17.2	Function Documentation	29
4.17.2.1	ViewProfile()	29

Chapter 1

Class Index

1.1 Class List

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

Authentication.Authentication	5
CommandParser.CommandParser This class represents a the Command Parser	6
firebase_creds.firebase_class This class is used for initialilzing the Firebase API connection and contains methods for interaction	9

Chapter 2

File Index

2.1 File List

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

src/Social Media App/CommandParser/ CommandParser.py	
Module used to transition control between modules	13
src/Social Media App/CommandParser/ help.py	
Module that displays information about available commands and their usage	14
src/Social Media App/Firebase/ Authentication.py	
Authentication module which prompts the user to login/sign_up	14
src/Social Media App/Firebase/ firebase_creds.py	
Used for initializing a database instance/connector with Firebase API credentials	15
src/Social Media App/Posts/ AddPost.py	
Used for adding Posts by the specified user	15
src/Social Media App/Posts/ DeletePost.py	
Used for Deleting a post	16
src/Social Media App/Posts/ EditPost.py	
Used for editing an existing post	17
src/Social Media App/Posts/ ViewPost.py	
Used for displaying posts data to the user	18
src/Social Media App/Profile/ AddFollowers.py	
Used for adding users to the following list of the logged in user	20
src/Social Media App/Profile/ DeleteAccount.py	
Used for Deleting account from Firebase database	21
src/Social Media App/Profile/ DeleteFollowers.py	
Used for deleting followers from the following list of the logged in user	22
src/Social Media App/Profile/ EditLocation.py	
Used for Updating Profile location information of the logged in user	23
src/Social Media App/Profile/ EditName.py	
Used for Updating Profile Name information of the logged in user	24
src/Social Media App/Profile/ QueryFollowers.py	
Used for Querying Followers list from the database	25
src/Social Media App/Profile/ QueryProfile.py	
Used for querying data for profile from Firebase	26
src/Social Media App/Profile/ ViewFollowers.py	
Used for Display/Print Followers on to the screen	27
src/Social Media App/Profile/ ViewProfile.py	
Used for displaying/printing profile information to the screen in a formatted way	29

Chapter 3

Class Documentation

3.1 Authentication.Authentication Class Reference

Public Member Functions

- def `__init__` (self, firebase, db)
Constructor accepts 2 parameters.
- def `login` (self)
Login method.
- def `password_reset` (self)
Password reset.
- def `register` (self)
Register method.

Public Attributes

- **firebase**
- **db**
- **timeout**

3.1.1 Constructor & Destructor Documentation

3.1.1.1 `__init__`()

```
def Authentication.Authentication.__init__ (
    self,
    firebase,
    db )
```

Constructor accepts 2 parameters.

Parameters

<i>firebase</i>	firebase object instance in order to execute commands to the Firebase console via Pyrebase API.
<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API

3.1.2 Member Function Documentation

3.1.2.1 login()

```
def Authentication.Authentication.login (
    self )
```

Login method.

Authenticates the user credentials with that found in the Firebase console

3.1.2.2 password_reset()

```
def Authentication.Authentication.password_reset (
    self )
```

Password reset.

Given a valid account email, sends a reset request via email

3.1.2.3 register()

```
def Authentication.Authentication.register (
    self )
```

Register method.

Registers a new account provided the username and password

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

- src/Social Media App/Firebase/[Authentication.py](#)

3.2 CommandParser.CommandParser Class Reference

This class represents a the Command Parser.

Public Member Functions

- def `__init__` (self, running, firebase, db)
Constructor accepts 3 parameters.
- def `parseCommand` (self, command)
Main method used to execute the parsing of commands.
- def `post_commands` (self, command_action, command_context)
Main method used to execute the parsing of commands.
- def `profile_commands` (self, command_action, command_context)
Parses Profile commands.
- def `clear` (self)
Clears the terminal after each command executes.

Public Attributes

- **running**
- **firebase**
- **db**
- **user_token**

3.2.1 Detailed Description

This class represents a the Command Parser.

suite of functions that are able to parse/execute commands

3.2.2 Constructor & Destructor Documentation

3.2.2.1 `__init__()`

```
def CommandParser.CommandParser.__init__ (
    self,
    running,
    firebase,
    db )
```

Constructor accepts 3 parameters.

Parameters

<i>running</i>	Boolean value that controls whole program should exit/run.
<i>firebase</i>	firebase object instance to execute commands via Pyrebase API.
<i>db</i>	firebase database object instance used to execute database commands

3.2.3 Member Function Documentation

3.2.3.1 parseCommand()

```
def CommandParser.CommandParser.parseCommand (
    self,
    command )
```

Main method used to execute the parsing of commands.

Parameters

<i>command</i>	string seperated by commas used to represent a command.
----------------	---------------------------------------------------------

3.2.3.2 post_commands()

```
def CommandParser.CommandParser.post_commands (
    self,
    command_action,
    command_context )
```

Main method used to execute the parsing of commands.

Parses Post commands (ex; add, view, delete) with context

Parameters

<i>command_action</i>	string that reps type of post command
<i>command_context</i>	string that represents the context(ex; username).

3.2.3.3 profile_commands()

```
def CommandParser.CommandParser.profile_commands (
    self,
    command_action,
    command_context )
```

Parses Profile commands.

Parses Profile commands (ex; add, view, delete) with context

Parameters

<code>command_action</code>	string that represents type of profile command
<code>command_context</code>	string that represents the context(ie; username).

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

- `src/Social Media App/CommandParser/CommandParser.py`

3.3 `firebase_creds.firebase_class` Class Reference

This class is used for initialilzing the Firebase API connection and contains methods for interaction.

Public Member Functions

- `def __init__ (self)`
Constructor.
- `def get_db (self)`
Getter method to retrieve DB instance.
- `def sign_in (self, email, password)`
Setter method sign-in a user and update the user-field.
- `def sign_up (self, email, password)`
Setter method that creates an account for the user.
- `def get_user_instance (self)`
Getter method used for user instance.
- `def get_username (self)`
Getter method that retrieves the username.
- `def set_username (self, username)`
Setter method that updates the local username.
- `def get_UID (self)`
Getter method to retrieve UID.

Public Attributes

- `config`
- `firebase`
- `auth`
- `user`
- `username`
- `db`

3.3.1 Detailed Description

This class is used for initialilzing the Firebase API connection and contains methods for interaction.

Contains API setup and the various getter/setter methods to make changes to the DB during first-start.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 `__init__()`

```
def firebase_creds.firebase_class.__init__ (
    self )
```

Constructor.

Contains all the information that is needed to interface with the database

3.3.3 Member Function Documentation

3.3.3.1 `get_db()`

```
def firebase_creds.firebase_class.get_db (
    self )
```

Getter method to retrieve DB instance.

Gets reference to database

Returns

Returns a firebase DB object

3.3.3.2 `get_UID()`

```
def firebase_creds.firebase_class.get_UID (
    self )
```

Getter method to retrieve UID.

Gets the Unique userID if the user is authenticated.

Returns

Returns the UID string

3.3.3.3 get_user_instance()

```
def firebase_creds.firebase_class.get_user_instance (
    self )
```

Getter method used for user instance.

Useful for signing out and identifying the user-ID(UID) of a given user.

Returns

Returns String representing the UID of the current logged-in user. Otherwise, returns None

3.3.3.4 get_username()

```
def firebase_creds.firebase_class.get_username (
    self )
```

Getter method that retrieves the username.

Returns

Returns the username, if user field is populated ONCE user is authenticated. Otherwise, returns None

3.3.3.5 set_username()

```
def firebase_creds.firebase_class.set_username (
    self,
    username )
```

Setter method that updates the local username.

Parameters

<i>username</i>	string value representing username
-----------------	------------------------------------

Updates the local_username during authentication, to use with the rest of the app

3.3.3.6 sign_in()

```
def firebase_creds.firebase_class.sign_in (
    self,
    email,
    password )
```

Setter method sign-in a user and update the user-field.

Parameters

<i>email</i>	string value representing email address
<i>password</i>	string value representing password

Authenticates the user and creates a session.

3.3.3.7 sign_up()

```
def firebase_creds.firebase_class.sign_up (
    self,
    email,
    password )
```

Setter method that creates an account for the user.

Parameters

<i>email</i>	string value representing email address
<i>password</i>	string value representing password

Authenticates the user and creates a session.

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

- src/Social Media App/Firebase/[firebase_creds.py](#)

Chapter 4

File Documentation

4.1 src/Social Media App/CommandParser/CommandParser.py File Reference

module used to transition control between modules

Classes

- class [CommandParser.CommandParser](#)
This class represents a the Command Parser.

Variables

- string **CommandParser.exit_text**

4.1.1 Detailed Description

module used to transition control between modules

Author

Anando Zaman, Graeme Woods

Date

March 1,2021

Controller module used to parse commands/execute commands

4.1.2 Variable Documentation

4.3.1 Detailed Description

Authentication module which prompts the user to login/sign_up.

Author

Anando Zaman

Date

February 10, 2021

4.4 src/Social Media App/Firebase/firebase_creds.py File Reference

Used for initializing a database instance/connector with Firebase API credentials.

Classes

- class [firebase_creds.firebase_class](#)

This class is used for initialzing the Firebase API connection and contains methods for interaction.

4.4.1 Detailed Description

Used for initializing a database instance/connector with Firebase API credentials.

Author

Anando Zaman

Date

February 10, 2021

4.5 src/Social Media App/Posts/AddPost.py File Reference

Used for addingPosts by the specified user.

Functions

- def [AddPost.addPost](#) (db, username, content, user_token)
Adds a Post with at most 100 characters, for a valid username.

4.5.1 Detailed Description

Used for adding Posts by the specified user.

Author

Graeme Woods

This is a standalone function just for POC purposes. It will later be incorporated into a class for Posts

Date

Feb 21, 2021

4.5.2 Function Documentation

4.5.2.1 addPost()

```
def AddPost.addPost (
    db,
    username,
    content,
    user_token )
```

Adds a Post with at most 100 characters, for a valid username.

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>username</i>	string value representing username of the user who is posting
<i>content</i>	string value representing the users' post content
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner

Returns

Returns the post that was added.

4.6 src/Social Media App/Posts/DeletePost.py File Reference

Used for Deleting a post.

Functions

- def [DeletePost.DeletePost](#) (db, current_user, user_token, post_id)
Delete a current logged-in users' post by post_id, if exists.

4.6.1 Detailed Description

Used for Deleting a post.

Author

Yuvraj Randhawa

Date

4.6.2 Function Documentation

4.6.2.1 DeletePost()

```
def DeletePost.DeletePost (
    db,
    current_user,
    user_token,
    post_id )
```

Delete a current logged-in users' post by post_id, if exists.

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>current_user</i>	string value representing username of the user who is posting
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner
<i>post_id</i>	string value corresponding to the post that is to be deleted

Removes the post via post_id from the allposts → posts branch, & profile → username → name → posts branch

4.7 src/Social Media App/Posts/EditPost.py File Reference

Used for editing an existing post.

Functions

- def [EditPost.EditPost](#) (db, current_user, user_token, post_id, content)
Edit contents of an existing post, if it exists under their profile posts branch.

4.7.1 Detailed Description

Used for editing an existing post.

Author

Yuvraj Randhawa

Date

4.7.2 Function Documentation

4.7.2.1 EditPost()

```
def EditPost.EditPost (
    db,
    current_user,
    user_token,
    post_id,
    content )
```

Edit contents of an existing post, if it exists under their profile posts branch.

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>current_user</i>	string value representing username of the user who is posting
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner
<i>post_id</i>	string value corresponding to the post that is to be deleted

Edit the post via `post_id` from the `allposts` → `posts` branch, & `profile` → `username` → `name` → `posts` branch.

if the `post_id` DOES NOT EXIST under the `current_user`, then prints "invalid PostID"

4.8 src/Social Media App/Posts/ViewPost.py File Reference

Used for displaying posts data to the user.

Functions

- def [ViewPost.ViewPostsByUser](#) (db, username, user_token)

- Void function that displays the posts by a given user.*
 - def `ViewPost.ViewPostsAll` (db, user_token)
 - Void function that displays all posts.*
 - def `ViewPost.ViewPostsFollowing` (db, firebase, user_token)
 - Void function that displays all posts made by the people you follow.*
 - def `ViewPost.printPost` (time, post_id, content, username="")
 - def `ViewPost.splitContent` (content)

4.8.1 Detailed Description

Used for displaying posts data to the user.

Author

Yuvraj

Date

Feb 20,2021

4.8.2 Function Documentation

4.8.2.1 ViewPostsAll()

```
def ViewPost.ViewPostsAll (  
    db,  
    user_token )
```

Void function that displays all posts.

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner

4.8.2.2 ViewPostsByUser()

```
def ViewPost.ViewPostsByUser (  
    db,  
    username,  
    user_token )
```

Void function that displays the posts by a given user.

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>username</i>	string that represents the user to query posts from
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner

4.8.2.3 ViewPostsFollowing()

```
def ViewPost.ViewPostsFollowing (
    db,
    firebase,
    user_token )
```

Void function that displays all posts made by the people you follow.

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>username</i>	string that represents the user to query posts from
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner

4.9 src/Social Media App/Profile/AddFollowers.py File Reference

Used for adding users to the following list of the logged in user.

Functions

- def [AddFollowers.AddFollowers](#) (db, current_user, user_to_follow, user_token)

Void function that updates the users' followings list.

4.9.1 Detailed Description

Used for adding users to the following list of the logged in user.

Author

Anando Zaman

Date

Feb 22, 2021

4.9.2 Function Documentation

4.9.2.1 AddFollowers()

```
def AddFollowers.AddFollowers (
    db,
    current_user,
    user_to_follow,
    user_token )
```

Void function that updates the users' followings list.

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>current_user</i>	string that represents the current logged-in user
<i>user_to_follow</i>	string that represents the username of the person that the current_user wants to follow
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner

4.10 src/Social Media App/Profile/DeleteAccount.py File Reference

Used for Deleting account from Firebase database.

Functions

- def [DeleteAccount.DeleteAccount](#) (firebase, db, current_user, user_token, exit_text)
Void function used for Deleting account from Firebase database.

4.10.1 Detailed Description

Used for Deleting account from Firebase database.

Author

Anando Zaman

Date

Development stage: Still in-progress

4.10.2 Function Documentation

4.10.2.1 DeleteAccount()

```
def DeleteAccount.DeleteAccount (
    firebase,
    db,
    current_user,
    user_token,
    exit_text )
```

Void function used for Deleting account from Firebase database.

Parameters

<i>firebase</i>	firebase object instance in order to execute commands to the Firebase console via Pyrebase API.
<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>current_user</i>	string that represents the current logged-in user
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner

4.11 src/Social Media App/Profile/DeleteFollowers.py File Reference

Used for deleting followers from the following list of the logged in user.

Functions

- def [DeleteFollowers.DeleteFollowers](#) (db, current_user, user_to_remove, user_token)

4.11.1 Detailed Description

Used for deleting followers from the following list of the logged in user.

Author

Anando Zaman

Date

Feb 22, 2021

4.11.2 Function Documentation

4.11.2.1 DeleteFollowers()

```
def DeleteFollowers.DeleteFollowers (
    db,
    current_user,
    user_to_remove,
    user_token )
```

Void function that updates the users' followings list by deleting users

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>current_user</i>	string that represents the current logged-in user
<i>user_to_remove</i>	string that represents the user that we want removed from followers list
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner

4.12 src/Social Media App/Profile/EditLocation.py File Reference

Used for Updating Profile location information of the logged in user.

Functions

- def [EditLocation.EditLocation](#) (db, username, user_token, new_location)
Update profile location information.

4.12.1 Detailed Description

Used for Updating Profile location information of the logged in user.

Author

Graeme Woods

Date

4.12.2 Function Documentation

4.12.2.1 EditLocation()

```
def EditLocation.EditLocation (
    db,
    username,
    user_token,
    new_location )
```

Update profile location information.

if new_location parameter is empty or None, display error message

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>username</i>	string that represents the current logged-in user
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner
<i>new_location</i>	string that represents the location that is to be used to update

4.13 src/Social Media App/Profile/EditName.py File Reference

Used for Updating Profile Name information of the logged in user.

Functions

- def `EditName.EditName` (db, username, user_token, new_name)
Update Profile name information.

4.13.1 Detailed Description

Used for Updating Profile Name information of the logged in user.

Author

Yuvraj Randhawa

Date

4.13.2 Function Documentation

4.13.2.1 EditName()

```
def EditName.EditName (
    db,
    username,
    user_token,
    new_name )
```

Update Profile name information.

if new_name is empty or None, display error message

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>username</i>	string that represents the current logged-in user
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner
<i>new_name</i>	string used for updating old name to new name

4.14 src/Social Media App/Profile/QueryFollowers.py File Reference

Used for Querying Followers list from the database.

Functions

- def [QueryFollowers.QueryFollowing](#) (db, username, user_token)
Getter method to query a persons' following list.
- def [QueryFollowers.QueryFollowers](#) (db, username, user_token)
Getter method to query a persons' followers list.

4.14.1 Detailed Description

Used for Querying Followers list from the database.

Author

Anando Zaman

Date

Feb 22, 2021

4.14.2 Function Documentation

4.14.2.1 QueryFollowers()

```
def QueryFollowers.QueryFollowers (
    db,
    username,
    user_token )
```

Getter method to query a persons' followers list.

if username does not exist, return None, otherwise returns the followers-list.

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>username</i>	string that represents username of the person to retrieve followers list from
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner

Returns

Returns a list of users followers. If invalid user, return None

4.14.2.2 QueryFollowing()

```
def QueryFollowers.QueryFollowing (
    db,
    username,
    user_token )
```

Getter method to query a persons' following list.

if username does not exist, return None, otherwise returns the following-list.

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>username</i>	string that represents username of the person to retrieve following list from
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner

Returns

Returns a list of users the person is following. If invalid user, return None

4.15 src/Social Media App/Profile/QueryProfile.py File Reference

Used for querying data for profile from Firebase.

Functions

- `def QueryProfile.QueryProfile (db, username, user_token)`

Returns a dictionary containing information about a users' profile(if exists), otherwise None.

4.15.1 Detailed Description

Used for querying data for profile from Firebase.

Author

Anando Zaman

Date

Feb 26,2021

4.15.2 Function Documentation

4.15.2.1 QueryProfile()

```
def QueryProfile.QueryProfile (
    db,
    username,
    user_token )
```

Returns a dictionary containing information about a users' profile(if exists), otherwise None.

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>username</i>	string that represents username of the person to retrieve profile info from
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner

4.16 src/Social Media App/Profile/ViewFollowers.py File Reference

Used for Display/Print Followers on to the screen.

Functions

- def [ViewFollowers.ViewFollowing](#) (db, username, user_token)
- def [ViewFollowers.ViewFollowers](#) (db, username, user_token)

4.16.1 Detailed Description

Used for Display/Print Followers on to the screen.

Author

Anando Zaman & Yuvraj Randhawa

Date

Feb 22, 2021

4.16.2 Function Documentation**4.16.2.1 ViewFollowers()**

```
def ViewFollowers.ViewFollowers (
    db,
    username,
    user_token )
```

Prints the followers to the screen

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>username</i>	string that represents username of the person to retrieve followings list from
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner

4.16.2.2 ViewFollowing()

```
def ViewFollowers.ViewFollowing (
    db,
    username,
    user_token )
```

Prints the followings list to the screen

Parameters

<i>db</i>	firebase database object instance used to execute database related commands via the Pyrebase API
<i>username</i>	string that represents username of the person to retrieve followings list from
<i>user_token</i>	string value needed to retrieve information from database in an authenticated manner

4.17 src/Social Media App/Profile/ViewProfile.py File Reference

Used for displaying/printing profile information to the screen in a formatted way.

Functions

- def [ViewProfile.ViewProfile](#) (db, username, user_token)

4.17.1 Detailed Description

Used for displaying/printing profile information to the screen in a formatted way.

Author

Graeme Woods

Date

4.17.2 Function Documentation

4.17.2.1 ViewProfile()

```
def ViewProfile.ViewProfile (
    db,
    username,
    user_token )
```

Prints the Profile information to the console in a formatted way

Parameters

<i>db</i>	takes the database object instance
<i>username</i>	takes the username as a string
<i>user_token</i>	takes the token string for authenticated data retrieval purposes

Returns

None

Index

- `__init__`
 - `Authentication::Authentication`, 5
 - `CommandParser::CommandParser`, 7
 - `firebase_creds::firebase_class`, 10
- `AddFollowers`
 - `AddFollowers.py`, 21
- `AddFollowers.py`
 - `AddFollowers`, 21
- `addPost`
 - `AddPost.py`, 16
- `AddPost.py`
 - `addPost`, 16
- `Authentication.Authentication`, 5
- `Authentication::Authentication`
 - `__init__`, 5
 - `login`, 6
 - `password_reset`, 6
 - `register`, 6
- `CommandParser.CommandParser`, 6
- `CommandParser.py`
 - `exit_text`, 13
- `CommandParser::CommandParser`
 - `__init__`, 7
 - `parseCommand`, 8
 - `post_commands`, 8
 - `profile_commands`, 8
- `DeleteAccount`
 - `DeleteAccount.py`, 21
- `DeleteAccount.py`
 - `DeleteAccount`, 21
- `DeleteFollowers`
 - `DeleteFollowers.py`, 22
- `DeleteFollowers.py`
 - `DeleteFollowers`, 22
- `DeletePost`
 - `DeletePost.py`, 17
- `DeletePost.py`
 - `DeletePost`, 17
- `EditLocation`
 - `EditLocation.py`, 23
- `EditLocation.py`
 - `EditLocation`, 23
- `EditName`
 - `EditName.py`, 24
- `EditName.py`
 - `EditName`, 24
- `EditPost`
 - `EditPost.py`, 18
- `EditPost.py`
 - `EditPost`, 18
- `exit_text`
 - `CommandParser.py`, 13
- `firebase_creds.firebase_class`, 9
- `firebase_creds::firebase_class`
 - `__init__`, 10
 - `get_UID`, 10
 - `get_db`, 10
 - `get_user_instance`, 10
 - `get_username`, 11
 - `set_username`, 11
 - `sign_in`, 11
 - `sign_up`, 12
- `get_UID`
 - `firebase_creds::firebase_class`, 10
- `get_db`
 - `firebase_creds::firebase_class`, 10
- `get_user_instance`
 - `firebase_creds::firebase_class`, 10
- `get_username`
 - `firebase_creds::firebase_class`, 11
- `login`
 - `Authentication::Authentication`, 6
- `parseCommand`
 - `CommandParser::CommandParser`, 8
- `password_reset`
 - `Authentication::Authentication`, 6
- `post_commands`
 - `CommandParser::CommandParser`, 8
- `profile_commands`
 - `CommandParser::CommandParser`, 8
- `QueryFollowers`
 - `QueryFollowers.py`, 25
- `QueryFollowers.py`
 - `QueryFollowers`, 25
 - `QueryFollowing`, 26
- `QueryFollowing`
 - `QueryFollowers.py`, 26
- `QueryProfile`
 - `QueryProfile.py`, 27
- `QueryProfile.py`
 - `QueryProfile`, 27

- register
 - Authentication::Authentication, [6](#)
- set_username
 - firebase_creds::firebase_class, [11](#)
- sign_in
 - firebase_creds::firebase_class, [11](#)
- sign_up
 - firebase_creds::firebase_class, [12](#)
- src/Social Media App/CommandParser/CommandParser.py, [13](#)
- src/Social Media App/CommandParser/help.py, [14](#)
- src/Social Media App/Firebase/Authentication.py, [14](#)
- src/Social Media App/Firebase/firebase_creds.py, [15](#)
- src/Social Media App/Posts/AddPost.py, [15](#)
- src/Social Media App/Posts/DeletePost.py, [16](#)
- src/Social Media App/Posts/EditPost.py, [17](#)
- src/Social Media App/Posts/ViewPost.py, [18](#)
- src/Social Media App/Profile/AddFollowers.py, [20](#)
- src/Social Media App/Profile/DeleteAccount.py, [21](#)
- src/Social Media App/Profile/DeleteFollowers.py, [22](#)
- src/Social Media App/Profile/EditLocation.py, [23](#)
- src/Social Media App/Profile/EditName.py, [24](#)
- src/Social Media App/Profile/QueryFollowers.py, [25](#)
- src/Social Media App/Profile/QueryProfile.py, [26](#)
- src/Social Media App/Profile/ViewFollowers.py, [27](#)
- src/Social Media App/Profile/ViewProfile.py, [29](#)
- ViewFollowers
 - ViewFollowers.py, [28](#)
- ViewFollowers.py
 - ViewFollowers, [28](#)
 - ViewFollowing, [28](#)
- ViewFollowing
 - ViewFollowers.py, [28](#)
- ViewPost.py
 - ViewPostsAll, [19](#)
 - ViewPostsByUser, [19](#)
 - ViewPostsFollowing, [20](#)
- ViewPostsAll
 - ViewPost.py, [19](#)
- ViewPostsByUser
 - ViewPost.py, [19](#)
- ViewPostsFollowing
 - ViewPost.py, [20](#)
- ViewProfile
 - ViewProfile.py, [29](#)
- ViewProfile.py
 - ViewProfile, [29](#)