

validateFirebaseToken(id_token: str) -> dict

The **validateFirebaseToken** method plays a crucial role in ensuring the security and integrity of user authentication within the application. When a user logs in or performs an action that requires authentication, they provide a Firebase ID token as proof of their identity. This token needs to be validated to ensure that it is genuine and has not been tampered with. The method receives the Firebase ID token as input and returns user token information if the validation is successful, indicating that the user is authenticated and authorized to access protected resources.

Parameters:

- **id_token** (str): A string representing the Firebase authentication token obtained from the client. This token serves as proof of authentication and authorization for the user.

root(request: Request)

The **root** method is responsible for handling HTTP GET requests to the root URL of the web application. It serves as the main entry point for rendering the application's main HTML template (**main.html**) and populating it with user-specific data, such as user information and timeline tweets.

Parameters:

- **request** (Request): An object representing the HTTP request received from the client. It contains information such as request headers, cookies, and query parameters.

addFile(file: UploadFile)

The **addFile** method facilitates the storage of files, such as images, documents, or multimedia content, within the Cloud Storage infrastructure. It enables users to upload files from their local devices to the cloud, making them accessible from anywhere with an internet connection. This method is particularly useful for applications that rely on user-generated content or require persistent storage for media files.

Parameters:

- **file** (**UploadFile**): The file to be uploaded.

blobList(prefix: str)

The **blobList** method retrieves a list of blobs (files) stored within Cloud Storage that match a specified prefix. Blobs are binary large objects representing individual files or objects stored in cloud-based storage systems. By providing a prefix, users can filter and retrieve blobs that belong to a specific directory, share a common naming convention, or meet other criteria.

Parameters:

- **prefix (str):** Prefix for the blobs.

downloadBlob(filename: str)

The **downloadBlob** method retrieves the content of a blob (file) from Cloud Storage based on the provided filename. It allows users to download files stored in the cloud to their local devices or access the file content programmatically within the application.

Parameters:

- **filename (str):** The name of the file to download.

getUser(user_token: dict)

The **getUser** method retrieves user details from Firestore, a flexible, scalable database provided by Google Cloud Platform. It allows applications to fetch user-specific data, such as profile information, preferences, and settings, from the Firestore database based on the provided user token.

Parameters:

- **user_token (dict):** A dictionary containing authentication or identification details of the user. This token serves as a unique identifier for the user within the application's ecosystem.

timelineTweets(user_token: dict)

The **timelineTweets** method retrieves timeline tweets for the user based on the provided user token. It aggregates tweets from the user's followed accounts and presents them in chronological order, creating a personalized timeline of relevant content.

updateUserForm(request: Request)

The **updateUserForm** method handles HTTP POST requests for updating the username of the authenticated user. It processes form data submitted by the user to modify their username and stores the updated information in the Firestore database.

Parameters:

- **request** (Request): An object representing the HTTP request received from the client. It contains form data submitted by the user.

addTweet(request: Request)

The **addTweet** method handles HTTP POST requests for adding new tweets to the Firestore database. It processes form data submitted by the user, including the tweet text and optional file upload, and stores the tweet information in Firestore.

Parameters:

- **request** (Request): An object representing the HTTP request received from the client. It contains form data submitted by the user, including tweet text and optional file upload.

searchUsers(request: Request)

The **searchUsers** method handles HTTP GET requests to find users based on a provided username query. It retrieves user data from the Firestore database matching the specified username query and renders the main HTML template (**main.html**) with the search results and other user-specific information.

Parameters:

- **request** (Request): An object representing the HTTP request received from the client. It contains information such as request headers, cookies, and query parameters.

findTweets(request: Request)

The **findTweets** method handles HTTP GET requests to find tweets based on a provided tweet text query. It retrieves tweet data from the Firestore database matching the specified tweet text query and renders the main HTML template (**main.html**) with the search results and other user-specific information.

Parameters:

- **request** (Request): An object representing the HTTP request received from the client. It contains information such as request headers, cookies, and query parameters.

userProfile(request: Request, id: str)

The **userProfile** method handles HTTP GET requests to display the profile page of a specific user identified by their unique ID. It retrieves user profile data and recent tweets associated with the user from the Firestore database and renders the **account.html** template with the user's profile information, recent tweets, and following status.

Parameters:

- **request** (Request): An object representing the HTTP request received from the client. It contains information such as request headers, cookies, and query parameters.
- **id** (str): The unique identifier of the user whose profile is to be displayed.

follow(request: Request, id: str)

The **follow** method handles HTTP GET requests to allow the authenticated user to follow another user identified by their unique ID. It retrieves the necessary user and authentication information, updates the list of followed users in the Firestore database, and redirects the user to the profile page of the followed user.

Parameters:

- **request** (Request): An object representing the HTTP request received from the client. It contains information such as request headers, cookies, and query parameters.
- **id** (str): The unique identifier of the user to be followed.

unfollowAccount(request: Request, id: str)

The **unfollowAccount** method handles HTTP GET requests to allow the authenticated user to unfollow another user identified by their unique ID. It retrieves the necessary user and authentication information, updates the list of followed users in the Firestore database by removing the specified user, and redirects the user to the profile page of the unfollowed user.

Parameters:

- **request** (Request): An object representing the HTTP request received from the client. It contains information such as request headers, cookies, and query parameters.
- **id** (str): The unique identifier of the user to be unfollowed.

deleteTweet(request: Request, id: str)

The **deleteTweet** method handles HTTP GET requests to allow the authenticated user to delete a specific tweet identified by its unique ID. It retrieves the necessary user and authentication information, verifies if the authenticated user is the owner of the tweet, and deletes the tweet from the Firestore database if ownership is confirmed. Finally, it redirects the user to the homepage after the deletion.

Parameters:

- **request** (Request): An object representing the HTTP request received from the client. It contains information such as request headers, cookies, and query parameters.
- **id** (str): The unique identifier of the tweet to be deleted.

downloadFileHandler(request: Request)

The **downloadFileHandler** method handles HTTP POST requests to allow users to download a file from cloud storage. It retrieves the necessary authentication information, validates the user's authentication token, retrieves the filename from the request form data, fetches the corresponding file from cloud storage, and initiates the file download.

Parameters:

- **request** (Request): An object representing the HTTP request received from the client. It contains information such as request headers, cookies, and form data.

editTweet(request: Request, id: str)

The **editTweet** method handles HTTP GET requests to allow users to edit a specific tweet identified by its unique ID. It retrieves the necessary authentication information, validates the user's authentication token, retrieves the tweet data from the Firestore database based on the provided tweet ID, and renders the **edit.html** template with the tweet data for editing.

Parameters:

- **request** (Request): An object representing the HTTP request received from the client. It contains information such as request headers, cookies, and query parameters.
- **id** (str): The unique identifier of the tweet to be edited.

updateTweet(request: Request, id: str)

The **updateTweet** method handles HTTP POST requests to allow users to update the content of a specific tweet identified by its unique ID. It retrieves the necessary authentication information, validates the user's authentication token, processes the form data submitted by the user, updates the tweet content in the Firestore database, and redirects the user to the homepage after the update.

Parameters:

- **request** (Request): An object representing the HTTP request received from the client. It contains information such as request headers, cookies, and form data.
- **id** (str): The unique identifier of the tweet to be updated.