**Twitter Microservices API**

Created by: Tyler Bloom

**Users (users.py):**

* createUser(): Adds a new user to the database when passed in the proper fields of username, password, and email.

HTTP Method: POST

Endpoint: /users

JSON Example:

{

“username”: “test”,

“password”: “sample”,

“email”: sample@gmail.com

}

Response Codes: HTTP 201 (Created)

HTTP 400 (Bad Request)

* users\_all(): Returns the entire User table and their attributes. Requires no JSON body to pass just request the endpoint.

HTTP Method: GET

Endpoint: /userlist

Response Codes: HTTP 200 (OK)

HTTP 400 (Bad Request)

* followers\_all(): Returns the entire Followers table and their attributes. Requires no JSON body to pass just request the endpoint.

HTTP Method: GET

Endpoint: /followerlist

Response Codes: HTTP 200 (OK)

HTTP 400 (Bad Request)

* authenticateUser(): A user can be verified using user authentication with werkzeug.security. The required fields of the JSON body are username and password. Once passed, the requested password is checked against the database hash password.

HTTP Method: POST

Endpoint: /authenticate

JSON Example:

{

“username”: “test”,

“password”: “sample”

}

Response Codes: HTTP 200 (Authorized)

HTTP 401 (Unauthorized)

HTTP 400 (Bad Request)

* addFollower(): Add a follower to a user that is inserted into the Followers table. The required fields of the JSON body are username and follower. The username is the person to be followed.

HTTP Method: POST

Endpoint: /follow

JSON Example:

{

“username”: “personFollowed”,

“follower”: “personFollowing”

}

Response Codes: HTTP 200 (OK)

HTTP 400 (Bad Request)

* removeFollower(): Removes a follower from a user that is inserted into the Followers table. The required fields of the JSON body are username and follower. The username is the person to be unfollowed.

HTTP Method: POST

Endpoint: /unfollow

JSON Example:

{

“username”: “personFollowed”,

“follower”: “personUnfollowing”

}

Response Codes: HTTP 200 (OK)

HTTP 400 (Bad Request)

**Timelines (timeslines.py):**

* getUserTimeline(): Retrieves the timeline for a specific user that is passed. The timeline is retrieved from the Tweets table with the foreign key referencing username from the User table.

HTTP Method: GET

Endpoint: /userTimeline

JSON Example:

{

“username”: “test”

}

Response Codes: HTTP 400 (Bad Request)

* getPublicTimeline(): Retrieves the public timeline of all users tweets. The timeline is retrieved from the Tweets table. No JSON body is required.

HTTP Method: GET

Endpoint: /publicTimeline

Response Codes: HTTP 304 (Page not modified)

HTTP 400 (Bad Request)

* getHomeTimeline(): Retrieves the home timeline of a specific user. The timeline contains the tweets of the people followed by the user specified. Every table is required to retrieve this timeline. The JSON body of a username must be passed.

HTTP Method: GET

Endpoint: /homeTimeline

JSON Example:

{

“username”: “test”

}

Response Codes: HTTP 400 (Bad Request)

* postTweet(): Creates a new tweet within the Tweets table under the specified user. The required fields of the JSON body are text and author.\

HTTP Method: POST

Endpoint: /tweets

JSON Example:

{

“text”: “Sample text of a posted tweet”,

“author”: “test”

}

Response Codes: HTTP 201 (Created)

HTTP 400 (Bad Request)