Author: Tushar Malakar

Twitter API Documentation:

Distributed System and Cloud Computing

**Purpose:** a cloud API (Application Programming Interface) to provide services to users. Services like to create a tweet, read a tweet, update a tweet, a delete tweet, comment on a tweet, like and dislike a tweet. In addition, “sign in” and “sign up” services to create and log-in to authentication services.

Programming Language: I used “python3.7.3” with “flask1.1.1” and “PyJWT1.7.1” to generate endpoints endpoints.

**Data base**: I used NoSQL database, “Mongodb” to store user account and tweets. Database name is TWITTER. TWITTER database has two collection.

1. Users Collection: to store all user account with username and password.

A close up of a logo

Description automatically generated

1. Tweet Collection: to store all tweet with the actual tweet and its owner, likes with liker name and actual like sign and comments with commenter name and actual comment.

A screenshot of a social media post

Description automatically generated

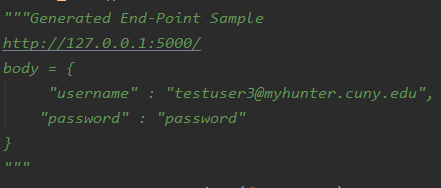
I have two types to services:

1. Public endpoints: where users do not need any authentication.
2. Private endpoints: where users need to authenticate himself through “create\_user” or “log\_in” endpoints. Note: I am using “json web token” as session cookie to provide authentication to authenticate a user.

Public Endpoints: Endpoints which do not need authentication. In this API services, I have few endpoints do not need to authentication. For example:

1. “create\_user” endpoint to create a new user with a HTTP POST request to server where user need to send a JSON body with request. This request will return, success message with a JWT session token or failure message with error message.

Note: user need to send this JWT session token to authenticate himself to use authenticated services.

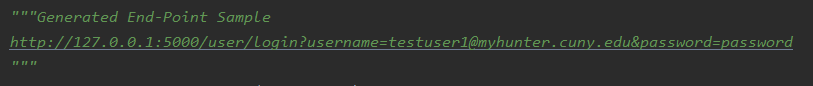


A screenshot of a cell phone

Description automatically generated

1. “log-in” endpoint to login to an account with a HTTP GET request to server. User should have a valid account with a valid username and a valid password to “log in.” This request will return, success message with a JWT session token or failure message with error message.

Note: user need to send this JWT session token to authenticate himself to use authenticated services.



A screenshot of a cell phone

Description automatically generated

1. “All\_tweet” endpoint to view all tweets with a HHTP GET request to server. This will return all active tweets.

A screenshot of a computer

Description automatically generated

Private Endpoints: To use these services user needs to add “session token” was return from “log in” or “sign up” to authenticate all these services. For example:

1. “create\_tweet” endpoint to create a new tweet with a HTTP POST request, in the tweet body requires a tweet and tweet owner. It will return success message on successful creation of tweet or failure message on failure to create a tweet.

A screenshot of a cell phone

Description automatically generated

1. “edit\_my\_tweet” endpoint to edit a tweet if she or he is the owner of the tweet with a HTTP POST request. In the body, “tweet ID” and “ownership” of tweet is required. It will return success message on successful edition of tweet or failure message on failure to edit a tweet.

A screenshot of a cell phone

Description automatically generated

1. “delete\_my\_tweet” endpoint to delete a tweet if she or he is the owner of the tweet with a HTTP POST request. In the body, “tweet ID” and “ownership” of tweet is required. It will return success message on successful deletion of tweet or failure message on failure to delete a tweet.

A screenshot of a cell phone

Description automatically generated

1. “my\_tweet” endpoint to see all my tweets with a HTTP GET request to server. In this

A screenshot of a cell phone

Description automatically generated

request “owner name” is required.