Team member CWID# Role

Alex Tran 891297442 Ops

Tu Tran Dev (Posting Microservices)

Joseph Hoang 889782900 Dev (Voting Microservices)

**Project Overview**

The scope of this project is to be able to implement two Microservices: Posting, Voting.

We will also need to be able to run multiple instances of each Microservices while using

load balancer to divide up the requests from each instance.

**Instructions to run**

**These are the library, and tools that will need to be install in order to run the project**

pip3 install --user Flask-API python-dotenv

pip3 install --user pugsql

sudo apt install --yes ruby-foreman

**First, we need to init the database for the both Posting and Voting Microservices**

flask init

**If you only run one instance per Microservices, please refresh multiple times.**

**Because of load balancer, it will take 1-2 to get to the correct $PORT**

**Then we will start the project by runing:**

foreman start

**Since, this the project required starting 3 instances of each Microservices.**

**Port number already been hardcoded in to work with only 3 or less instances.**

**If you want to run multiple instances, please use this link to run the project:**

foreman start -m web1=<number of instances => 3>,web2=<number of instances => 3>,caddy=1

**Flask Error**

If you get Error: No such command init, please create .env file

.env will contain the following {

FLASK\_APP=vote

FLASK\_APP=api

FLASK\_ENV=development

APP\_CONFIG=api.cfg

}

**Port In Use Error**

sudo lsof -i -P -n | grep LISTEN

kill <port id>

**Developer Ops**

For Ops role, I will be using Caddy for webserver and Gunicorn for WSGI server

User will be able to access to Post or Vote Microservices by using these hosts:

localhost:2015/post

localhost:2015/vote

**Voting Microservices**

View all votes:  [localhost:2015/vote/api/v1/resources/votes/all](http://127.0.0.1:5000/api/v1/resources/votes/all)

View votes by vote id:  [localhost:2015/vote/api/v1/resources/votebyid/1](http://127.0.0.1:5000/api/v1/resources/votebyid/1)

Upvote a post:  [localhost:2015/vote/api/v1/resources/upvote](http://127.0.0.1:5000/api/v1/resources/upvote)

Have to input in json format.

Example {“postID”: 10}

Downvote a post:  [localhost:2015/vote/api/v1/resources/downvote](http://127.0.0.1:5000/api/v1/resources/downvote)

Have to input in json format.

Example {“postID”: 11}

Report the number of upvotes and downvotes for a post:

localhost:2015/vote/api/v1/resources/votesbypostid/101

List the *n* top-scoring posts to any community:

localhost:2015/vote/api/v1/resources/toppostscore/3

Given a list of post identifiers, return the list sorted by score:

[localhost:2015/vote/api/v1/resources/listsortedbyscore](http://127.0.0.1:5000/api/v1/resources/listsortedbyscore)

Have to input a list: Example: [10, 13, 15]

**Posting Microservices**

For Posting Microservices, below is the following route to specific functions

To view home page for post:

REQUEST GET

localhost:2015/post

To view all posts:

REQUEST GET

localhost:2015/post/posts/all

To view a specific post:

REQUEST GET

localhost:2015/post/posts/<id>

example: localhost:2015/post/posts/1

To delete a specific post:

REQUEST GET, DELETE

localhost:2015/post/posts/delete/<id>

example: localhost:2015/post/posts/delete/1

To view a specific, create:

json format is required, all fields are required,

REQUEST GET, POST

localhost:2015/post/posts

example:

{

"Username": "User 1",

"PostTitle": "Post Title 1",

// "PostDate": "01/01/2020",

"Content": "Content 1",

"Community": "Community 1",

"URLResource": "www.URLResource1.com"

}

To view a specific community in posts:

REQUEST GET

http://localhost:5000/posts/<community name>

example: localhost:2015/post/posts/Community\_1