**Team member CWID# Role**

Alex Tran 891297442 Dev (Backend for Frontend)

Tu Tran 888750130 Dev (Posting Microservices)

Joseph Hoang 889782900 Dev (Voting Microservices)

**Project 2 Spring 2020**

**Project Overview**

Dev 1: Posting Microservices

* Create a new post
* Delete an existing post
* Retrieve an existing post
* List the n most recent posts to a particular community
* List the n most recent posts to any community

Dev 2: Voting Microservices

* Upvote a post
* Downvote a post
* Report the number of upvotes and downvotes for a post
* List the n top-scoring
* Given a list of post identifiers, return the list sorted by score

Dev 3: Aggregating posts and votes with a BFF

* By using BFF pattern, this microservices will be pulling data from the other services and expose it in a format of RSS that can be used directly by the frontend

For this project, each microservices will be given a default port to ensure that all microservices will work properly all

* Posting microservices: localhost:5000/post/posts/all
* Voting microservices: [localhost:5100/api/v1/resources/votes/all](http://127.0.0.1:5100/api/v1/resources/votes/all)
* BFF microservices: localhost:5200/recent

**Library Requirement**

**Please Run this line before installing Library**

* Install Flask

$ sudo apt install python3-flask

$ pip3 install Flask-API

$ pip3 install --user Flask-API python-dotenv

* Install foreman & Gunicorn3

$ sudo apt install --yes ruby-foreman

$ sudo apt install --yes gunicorn3

Dev 1: Library

* Update list of software source

$ sudo sed -i -re \

's/([a-z]{2}\.)?archive.ubuntu.com|security.ubuntu.com/old-releases.ubuntu.com/g' \

/etc/apt/sources.list

* Install AWS CLI

$ sudo apt update

$ sudo apt install --yes awscli

* Configure AWS CLI

$ aws configure

AWS Access Key ID [None]: **fakeMyKeyId**

AWS Secret Access Key [None]: **fakeSecretAccessKey**

Default region name [None]: **us-west-2**

Default output format [None]: **table**

* Install Boto 3

$ sudo apt install --yes python3-boto3

* Install DynamoDB Flask extension

$ pip3 install --user flask-dynamo

* Check python 3
  + $ which flask

Follow your direction, open the flask file

The first line should end with “python3”.

If it ends with “python”, make it “python3” and save the file

Example:

Incorrect: #!/usr/bin/python

Correct: #!/usr/bin/python3

Dev 2: Library

* Installing Redis

$ sudo apt update

$ sudo apt install --yes redis python3-hiredis

* You can verify that Redis is up and running with:

$ redis-cli ping

Server will respone: PONG

* Installing the high-speed Python library for Redis

$ sudo apt install --yes python3-hiredis

* Install the Flask-And-Redis extension

$ pip3 install --user Flask-and-Redis

Dev 3: Library

* Install Request

$ pip3 install --user requests

* Install Feed Generator

$ sudo apt update

$ sudo apt install --yes python3-lxml

$ pip3 install --user feedgen

**Instruction to run**

* Start the three microservices

$foreman start

* Initiate the database on a second terminal.
* First. Init dev 1 database, then init dev 2 database

$ flask init

$ export FLASK\_APP=redisvotes.py

$ flask initredis

Posting Microservices

http://localhost:5000

* Display all posts:
  + <http://localhost:5000/posts/all>
* Delete post by PostID:
  + [http://localhost:5000/posts/delete/<int:PostID>](http://localhost:5000/posts/delete/%3cint)

Example:

* + - <http://localhost:5000/posts/delete/100>
* Display post by PostID:
  + [http://localhost:5000/posts/<int:PostID>](http://localhost:5000/posts/%3cint%3e)

Example

* + - <http://localhost:5000/posts/10>
* Display default 2 recent posts with create post:
  + <http://localhost:5000/posts>

To create a new post, please copy this Json then change value if needed

{"Username": "User 100", "PostTitle": "Post Title 100", "Content": "Content 100", "Community": "home", "URLResource": "www.URLResource100.com"}

* Display n recent post with create post:

Example

n = 15 (can be changed to any value)

* + <http://localhost:5000/posts?n=15>

To create a new post, please copy Json above then change the values if needed

* Display n most recent post in school community:
  + [http://localhost:5000/posts/<string:Community>/<int:n](http://localhost:5000/posts/%3cstring:Community%3e/%3cint:n)>

Example

* + - <http://localhost:5000/posts/school/5>

Voting Microservices

* View votes by postID:
  + http://127.0.0.1:5100/votebypostid/<postID>

Examle: http://127.0.0.1:5100/votebypostid/1

* View all votes:
  + http://127.0.0.1:5100/allvotes

Examle: http://127.0.0.1:5100/allvotes

* View 1 votes by vote id:
  + http://127.0.0.1:5100/vote/<voteID>

Examle: http://127.0.0.1:5100/vote/1

* Upvote a post:
  + http://127.0.0.1:5100/upvote/<postID>
  + Have to input in json format. Example: {“postID”: 2}
* Downvote a post:
  + http://127.0.0.1:5100/ downvote /<postID>
  + Have to input in json format. Example: {“postID”: 2}
* List the n top-scoring posts:
  + http://127.0.0.1:5100/toppostscore/<number>
  + Example: http://127.0.0.1:5100/toppostscore/10
* Given a list of post identifiers, return the list sorted by score:
  + http://127.0.0.1:5100/listsortedbyscore
  + Have to input in JSON: Example: {"listPostID": [1, 2, 25]}

RSS feeds provided by the BFF Microservices

* The 25 most recent posts to any community
  + <http://localhost:5200/recent>
* The 25 most recent posts to a particular community
  + [http://localhost:5200/<string>/<int>](http://localhost:5200/%3cstring%3e/%3cint%3e)
  + <http://localhost:5200/school/25>
* Top 25 posts to any community, sorted by score
  + <http://localhost:5200/score>
* The top 25 posts to a particular community, sorted by score
  + [http://localhost:5200/score/<string>](http://localhost:5200/score/%3cstring%3e%20_2)
  + <http://localhost:5200/score/school>
* The hot 25 posts to any community, ranked using Reddit hot ranking algorithm
  + <http://localhost:5200/hot>