Scalable Computing - CS7NS1
Supplemental Task

VARNIT GOEL | 17313068

August 17, 2018

1 Question:

You are required to deliver a REST service, that provides an interface for submitting a set of GitHub repositories (identified as a list of strings of the form "username/repositoryname"). You service should identify the set of GitHub developers who have contributed to any of these repositories in 2018, and for this user set, provide an ordering that ranks these users according to each of the following criteria:

1. Total number of commit contributions to any project to which a user has a contributed.
2. Total number of commit contributions as above, but restricted to projects that are members of the original submitted set.
3. The number of known programming languages for each user (presuming that the languages of any repository committed to are known to the user)
4. The weekly commit rate of users (provide a weekly rank ordering) for the submitted project set, for 2018.
5. The average commit rate of each user to any project, for 2018.
6. The total number of collaborators in 2018 (ie. a count of other users who have contributed to any project that the user has contributed to).

Your rest service should email the results of calculation to the submitter once complete.

2 Introduction

Python programming language is used to execute the assignment where, we're using FLASK library/class to build the RESTFull API for our program. Flask is a Python-based micro-framework that enables you to quickly build web applications. REST, REpresentational State Transfer is an architectural style, and an approach to communications that is often used in the development of Web services. The application gives information about the users, the commits in a specific repository, weekly & average commits, languages used by the user, and the total number of collaborators in 2018 etc. and it emails the results to the user.

How the application is working:
The application takes input from the user including,

1. Number of users
2. Username of the user
3. Repository name of the user

and gives the respective output according to the each given criteria.

The link to the repository is:
https://github.com/varnitgoel/REST-SERVICE-API—CS7NS1

The name of the repository is:
REST-SERVICE-API—CS7NS1

3 Development of application

3.1 Commands/Keywords/Libraries/Classes used in the application

GET - /api/Category - Retrieve all categories/comments
POST - /api/Category - Add a new category/comment

The use of 'GET' command in the declaration of each criteria: The browser tells the server to just get the information stored on that page and send it. This is the most common method.

The function **(@app.route('/'))** is given a name which is also used to generate URLs for that particular function, and returns the message we want to display in the user's browser.

Flask - This is a micro-framework for Python flask_restful - This is an extension for Flask that adds support for quickly building of REST APIś.

Nested_lookup: A small Python library which enables key lookups on deeply nested documents. Documents may be built out of dictionaries (dicts) and/or lists.

For the E-Mail function, we've used SMTPLIB and SMTPException: It defines a Simple Mail Transfer Protocol client session object that can be used to send mail to any Internet machine

Throughout the programming of this application/program, we've used GitHub API's available freely over the internet and easy to implement or merge in your program. To run those API's, user input is taken and put (append) in individual dictionaries.Rest, my coding is pretty straightforward with comments mentioned wherever necessary.
The URL for all the GitHub API's used is mentioned along with all the problems I encountered while programming this application.These were solved by the references mentioned in the reference section below.

To run the application
Anyone can clone the repo and run it on their respective systems on the local host
$set FLASK_APP=main.py
$flask run
Running on http://127.0.0.1:9619/

If the server doesn't start, use:
flask run –host=0.0.0.0 (in the command line)

References

[1] http://flask.pocoo.org/docs/0.12/quickstart/

[2] https://github.com/dustin/py-github

[3] https://stackoverflow.com/questions/13081532/return-json-response-from-flask-view

[4] https://developer.github.com/v3/repos/commits/#list-commits-on-a-repository

[5] https://developer.github.com/v3/repos/statistics/#get-contributors-list-with-additions-deletions-and-commit-counts

[6] https://pythonspot.com/flask-web-app-with-python/

[7] https://pythonspot.com/flask-web-app-with-python/

[8] https://blog.spritecloud.com/2010/03/creating-and-sending-html-e-mails-with-python/

[9] https://github.com/russellballestrini/nested-lookup

[10] https://stackoverflow.com/questions/19595702/using-html-templates-to-send-emails-in-python