Dream Broker Programming Assignment: Instruction for Checking the program

Prepared by: Humayun Rashid

Email: Humayun.rashid@tuni.fi

The application has been developed using python in windows 10. It has been tested with Postman and curl. The test code has also been installed using Python. The server has been deployed to Heroku and testing code has been integrated with Travis CI. Also, the zip file contains all the files that can be run from local server with following instructions.

GIT Link: https://github.com/humayun-rashid/Dream-Broker_Assignment

Heroku Link: https://dream-broker-assignment.herokuapp.com/analyze

Travis CI : https://travis-ci.com/humayun-rashid/Dream-Broker_Assignment

Details instruction has been given below.

Installation:

Python 3.7.2 : https://www.python.org/downloads/release/python-372/

Curl: https://curl.haxx.se/windows/

Postman: https://www.getpostman.com/downloads/

Modules:

Several modules need to be installed to run the code properly. It is strongly recommended to install the libraries using pip.

PIP: https://pip.pypa.io/en/stable/installing/

After importing pip, it is important to add the environment path url. libraries can be installed using pip from command. Install following modules using pip:

Flask: pip install flask

Requests: pip install requests

unittest: pip install unittest2

Execution:

If Python has been installed properly with run time environment, the code can be executed from command window.

```
Command Prompt - python  "C:\Users\H. Rashid Raahat\Desktop\Dream Broker Assignment\app.py"
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\H. Rashid Raahat>python "C:\Users\H. Rashid Raahat\Desktop\Dream Broker Assignment\app.py"
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 886-158-615
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Functionality can be checked through curl command and Postman.

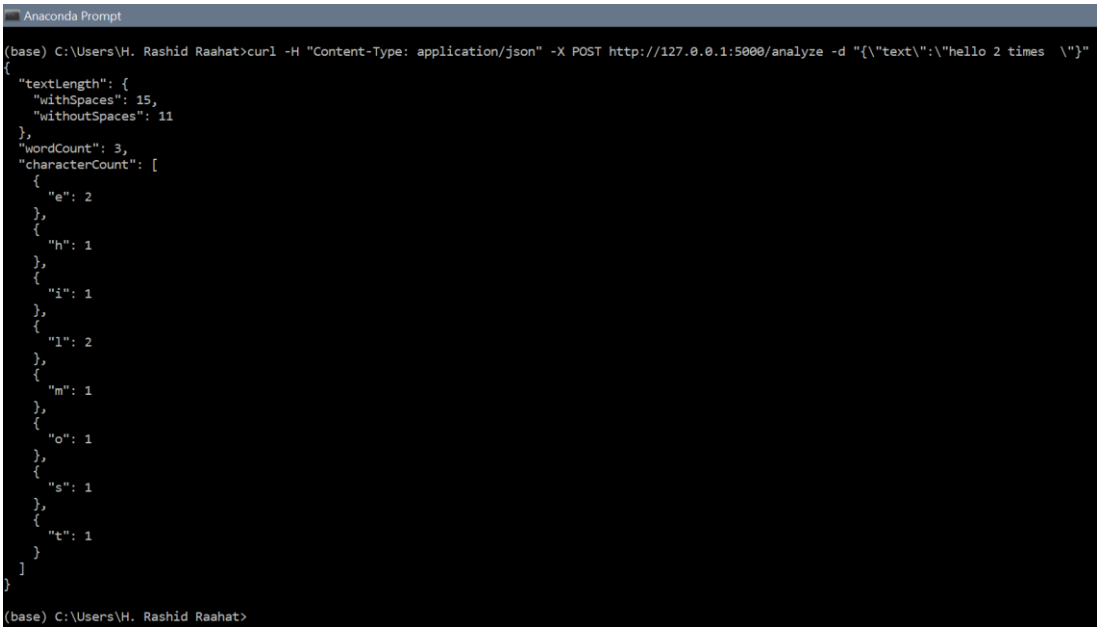
Curl:

It needs to be taken in attention that curl command is different for different OS and different software version. My one is for windows and can be tested with following provided command.

Command: (copy and paste should work if curl is properly installed)

curl -H "Content-Type: application/json" -X POST http://127.0.0.1:5000/analyze -d "{\"text\":\"hello 2 times \"}"

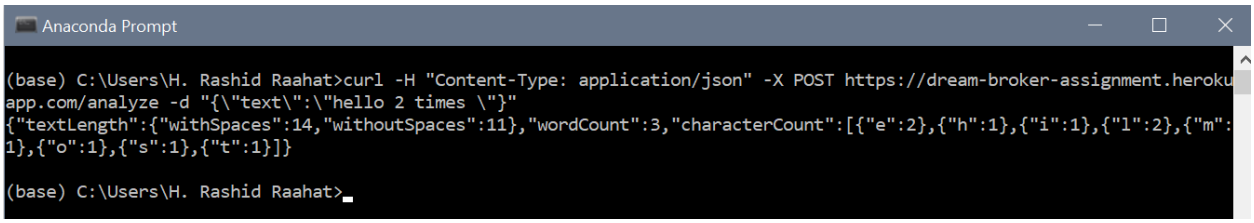
Response:



For Heroku deployment check, please use following command:

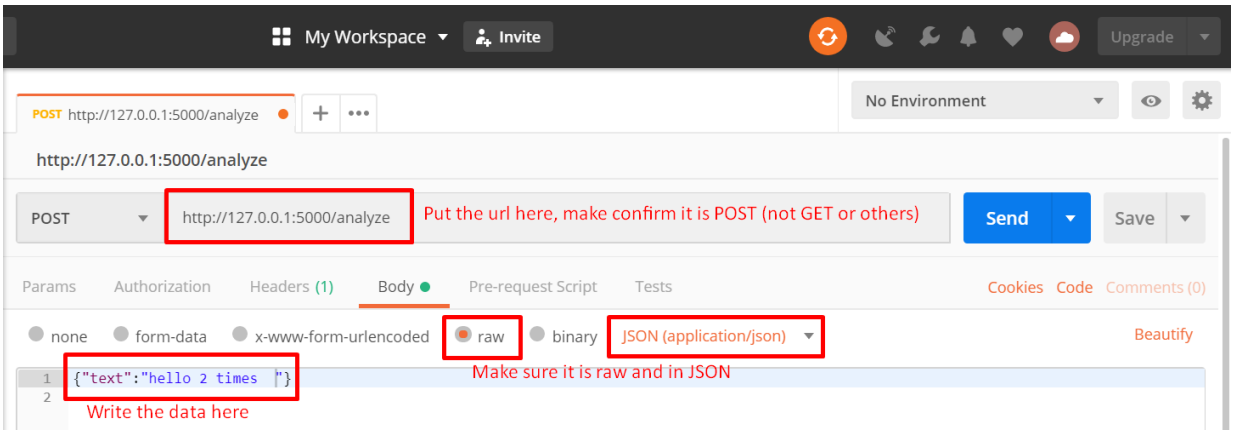
curl -H "Content-Type: application/json" -X POST https://dream-broker-assignment.herokuapp.com/analyze -d "{\"text\":\"hello 2 times \"}"

it will return following response:

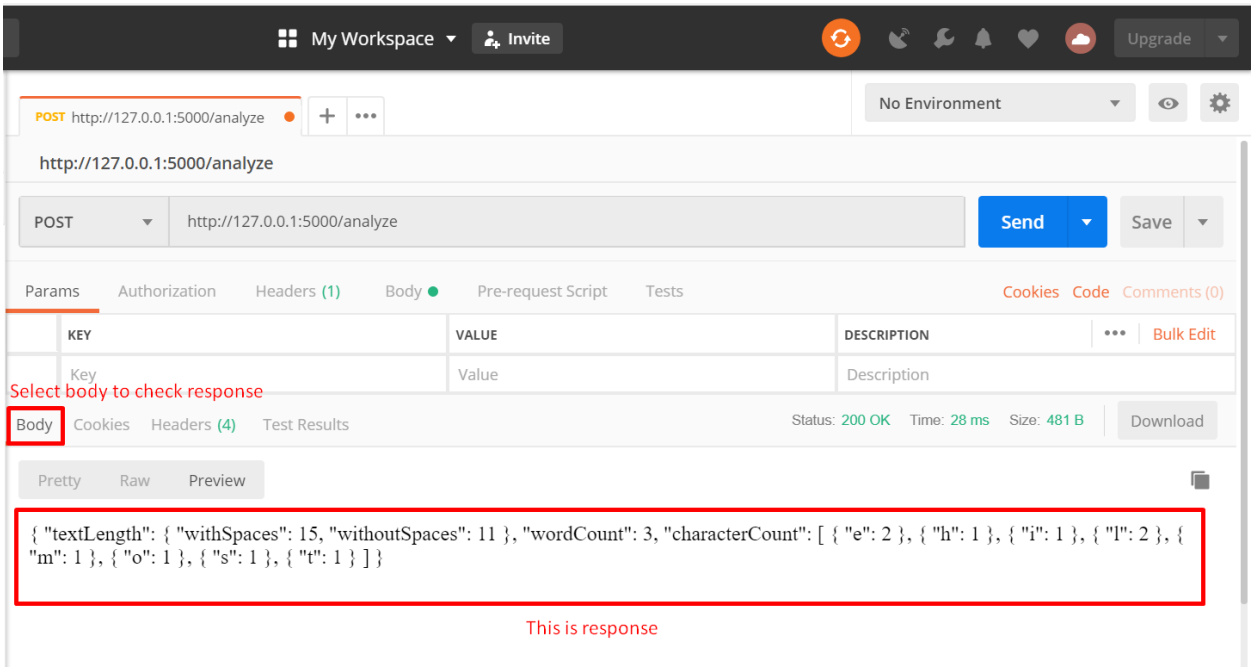


Using Postman:

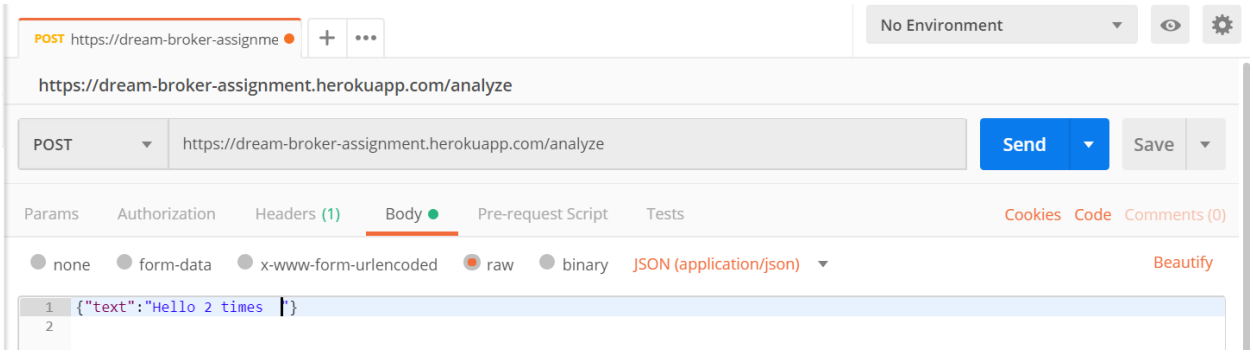
It is very simple to use Postman. Although, this was not asked in the task, but I always prefer to test any API with postman. The setup needs to be like following.



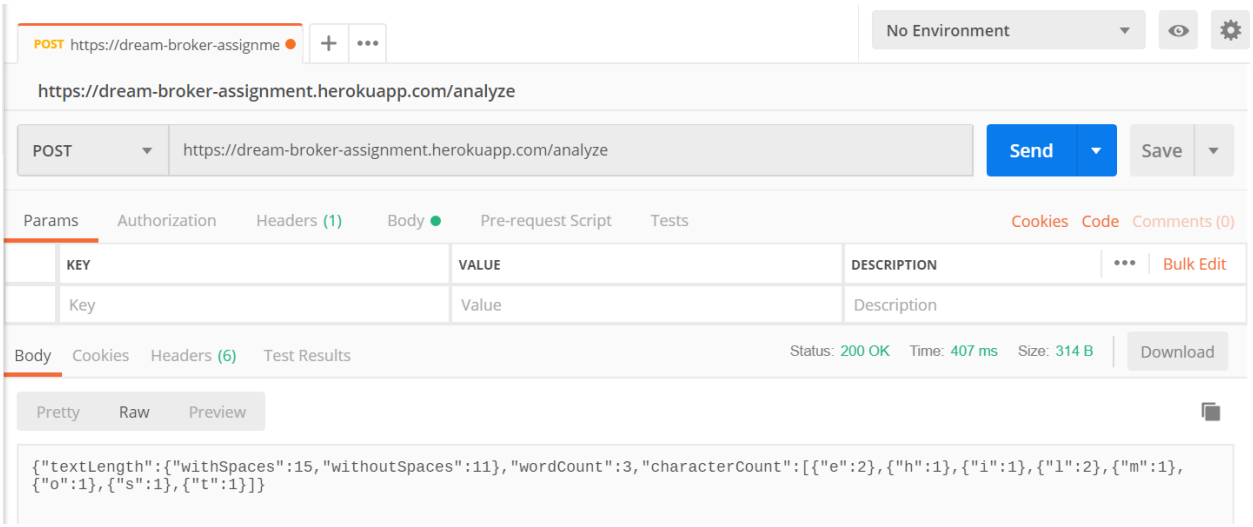
Response:



For Heroku-



Response is -



Testing:

Simple unit testing has been done. The test file can be found in the zip folder. There are two test files test.py and another is test_(local_server).py. test.py can be used to test Heroku deployment. test_(local_server).py can be used to test local server. Following this functions/method has been tested:

1. If post request returns the expected result
2. Text length counting test
3. Character counting test
4. Word counting test

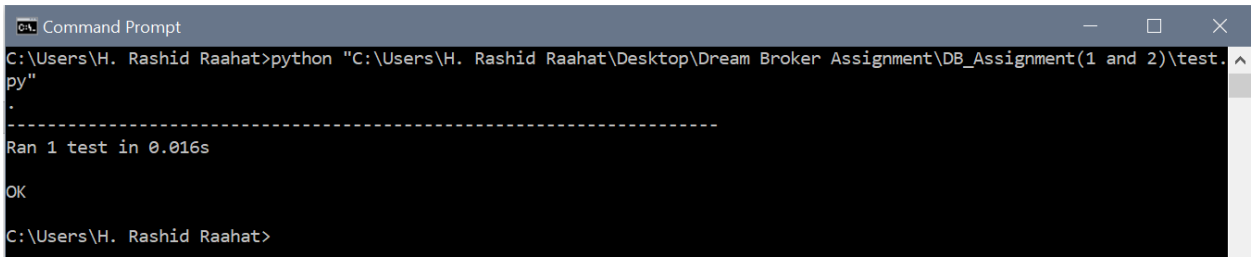
For checking, local server URL need to be replaced carefully in test_(local_server).py. For my case, server url is: http://127.0.0.1:5000

so, in test_(local_server).py, I used : http://127.0.0.1:5000/analyze in a varable server_url.

For testing, it needs to be replaced with proper localhost url. (Example: http://localhost:8000/analyze)

```
C:\Users\H. Rashid Raahat>python "C:\Users\H. Rashid Raahat\Desktop\Dream Broker Assignment\app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 886-158-615
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

It can be run from command if Python and run time environment has been properly configured. If testcase passes, output will be similar like this:



Travis CI has been used for continuous test deployment. For this, .travis.yml file has been created and uploaded to the GIT. IT can be checked from following link:

https://travis-ci.com/humayun-rashid/Dream-Broker_Assignment