nyctaxiserver

Description

A python (flask-restplus) based Web API for returning statistics related to trips made by new york taxis based on Google BigQuery Data.

Steps to run

In cass you face issues please run the below commands in sudo/admin privileges

1. *Python 3.5 or above and associated pip* should be installed on the system (the code has been verified on *Python 3.5.2*). Following commands may be used if *Python 3.5 or above and associated pip* is not installed on your machine

```
         -- Linux (verfied on Ubuntu 16.04.6 LTS):
             test@testmachine:~/test/repodir$ sudo apt-get install
python3
             test@testmachine:~/test/repodir$ sudo apt-get install
python3-pip
```

2. **Virtual Environment Setup** : (Required Only Once. If setup has been completed earlier, please go to Step 3)

```
 a. You may *install/configure* a virtual environment by running
commands from bash shell as follows. (*Note* :In case ``python``
command  on your machine points to  *Python 3+* environment, replace
``python3``  with ``python`` in below commands)

     -- Linux (verfied on Ubuntu 16.04.6 LTS)

         test@testmachine:~/test/repodir$ cd nyctaxiserver
         test@testmachine:~/test/repodir/nyctaxiserver$ python3 -m pip
install --user virtualenv
         test@testmachine:~/test/repodir/nyctaxiserver$ python3 -m
virtualenv venvtaxiapi
         test@testmachine:~/test/repodir/nyctaxiserver$ source
venvtaxiapi/bin/activate
         (venvtaxiapi) test@testmachine:~/test/repodir/nyctaxiserver$
python -m pip install -r requirements.txt

     You should see the terminal as below after above commands

         (venvtaxiapi)
test@testmachine:~/testdir/repodir/nyctaxiserver$
```

```
 b.   You may also setup the environment using following commands from
bash shell. You will need navigate (*cd*) to the folder
``nyctaxiserver`` (the one that contains the file *README.md* and
folder *app* ) and then run below commands to install and activate the
virtual environment in which the API server would run. (*Note* : In
case ``python`` command  on your machine points to  *Python 3+*
environment, please  replace ``python3`` with ``python`` in the
`envsetup.sh` file)

     -- Linux (verfied on Ubuntu 16.04.6 LTS):
         test@testmachine:~/test/repodir$ cd nyctaxiserver
         test@testmachine:~/test/repodir$ chmod a+x envsetup.sh
         test@testmachine:~/test/repodir/nyctaxiserver$ source
envsetup.sh

     You should see the terminal as below after above commands

         (venvtaxiapi)
test@testmachine:~/testdir/repodir/nyctaxiserver$
```

3. **Activate Virtual Environment** : Do this step if conifguration (Step 2) has already been setup and you want to start the API Server. You may ignore this step if coming from Step 2, as Step 2 should have already started the Server**

   Navigate (*cd*) to the folder *nyctaxiserver* folder and run the following commands to run the API server

```
 -- Linux (verfied on Ubuntu 16.04.6 LTS):

     test@testmachine:~/test/repodir$ cd nyctaxiserver
     test@testmachine:~/test/repodir/nyctaxiserver$ source
venvtaxiapi/bin/activate/

 You should see the terminal as below after above commands

     (venvtaxiapi) test@testmachine:~/testdir/repodir/nyctaxiserver$
```

4. **Start Server** From the bash cell (or command line) by running the following command. The virtual environment should already be active based on Steps 3 or 4.

```
 -- Linux (verfied on Ubuntu 16.04.6 LTS):
     (venvtaxiapi) test@testmachine:~/testdir/repodir/nyctaxiserver$
python runserver.py
```

   In case this step is successfull you should see the server running at 'http://localhost:5000/' (you can change port number using Step 7)

5. **Google S2Geometry Build** : In case running the above command displays `Could not load google S2Geometry will be using s2sphere for s2id computation` then you will have

to build the S2Geometry library on your system in case you want S2IDs based on S2Geometry. To do this you should to either of the following steps

-- **Install using the script provided** : Navigate to the folder `make2idpackage` present at `test@testmachine:~/test/repodir/nyctaxiserver/make2idpackage/$` and run the command `./make_s2python_package.sh`. The following steps should achieve it (**sudo** privileges are required so you may have to enter *root password* when asked)

```
     -- Linux (verfied on Ubuntu 16.04.6 LTS):
          (venvtaxiapi)
test@testmachine:~/testdir/repodir/nyctaxiserver$ cd makes2idpackage
          (venvtaxiapi)
test@testmachine:~/testdir/repodir/nyctaxiserver/makes2idpackage$
chmod a+x make_s2python_package.sh
          (venvtaxiapi)
test@testmachine:~/testdir/repodir/nyctaxiserver/makes2idpackage$
./make_s2python_package.sh
          (venvtaxiapi)
test@testmachine:~/testdir/repodir/nyctaxiserver/makes2idpackage$ cd
..
          (venvtaxiapi)
test@testmachine:~/testdir/repodir/nyctaxiserver$ python runserver.py

     Please Make sure that  `(venvtaxiapi)` is actcvated in the shell.
If not activate it and then run the server. Following should be the
sequence then.

          (venvtaxiapi)
test@testmachine:~/testdir/repodir/nyctaxiserver$ cd makes2idpackage
          (venvtaxiapi)
test@testmachine:~/testdir/repodir/nyctaxiserver/makes2idpackage$
chmod a+x make_s2python_package.sh
          (venvtaxiapi)
test@testmachine:~/testdir/repodir/nyctaxiserver/makes2idpackage$
./make_s2python_package.sh

test@testmachine:~/testdir/repodir/nyctaxiserver/makes2idpackage$ cd
..
         test@testmachine:~/test/repodir/nyctaxiserver$ source
venvtaxiapi/bin/activate/
          (venvtaxiapi)
test@testmachine:~/testdir/repodir/nyctaxiserver$ python runserver.py
```

-- **Install using offficial documentation** : Build based on steps present at http://s2geometry.io/about/platforms.html. Then you will need to copy the files `pywraps2.py` and `_pywraps2.so` files from folder `test@testmachine:~/test/<somefolderonyourmachine>/s2geometry/build/python$` to `test@testmachine:~/test/repodir/nyctaxiserver/app/main/utils/S2Lib$`

P.S: This library does officia support *Python3+* yet so there even after a local build, there may be issues loading the library.

6. **Google Service Account Configuration** : In case you need to use your own google service account (and not the default which I have provided) follow the below steps else go to Step 3.

   a. Login to your google account and follow the steps mentioned at https://support.google.com/a/answer/7378726?hl=en. It will download your service account key information as a `.json` file to your machine

   b. Replace the contents of the file `bqconfig.json` file present at location `test@testmachine:~/test/repodir/nyctaxiserver/app/configuration/` with the contents of `.json` file that was downloaded to your machine in previous Step (i.e. 2a)

   c. Open `bqconfig.json` and copy the value for the key `"project_id"`. For example in case your `bqconfig.json` contents look like below you need to copy `"ABC123"`. Close the file `bqconfig.json`

   ```
   {
       "type": "service_account",
       "project_id": "ABC:123",
       ...
   ```

   d. Open `dbonfig.py` present at location `test@testmachine:~/test/repodir/nyctaxiserver/app/configuration/` and set the value of the key **SVC_ACCNT_PROJECT_NAME** of dictionary **DATABASE_CONFIG** to the value copied in prevous Step 2c. (i.e `"ABC:123"` as based on above example) . Below is an illustration on how the file `dbonfig.py` should look after changes. Save and close `dbconfig.py` after making changes.

   ```
   DATABASE_CONFIG = {

                           'big_query_keys':{
                               'bqconfig':os.path.join
   (configdir,'bqconfig.json'),
                               'yourkey':os.path.join
   (configdir,'enteryourkey.json'),

                           },
                           'SVC_ACCNT_PROJECT_NAME' : "ABC:123",
                           ....
                           ....
   ```

7. **Project Configurations** : You may configure following items are per your conbenience (the project should work with default configurations also if no port conflicts are there)

a. **Port Number** : In case you do not want to use the default port number, open the file `config.py` at location `test@testmachine:~/test/repodir/nyctaxiserver/app/configuration/` and set the value of the variable **PORT** to the value to the values you desire. Foe example if you want to use the port number 9000 the file `config.py` should look like below. Save and close the file `config.py` after making changes.

```
 import  os
 configdir = os.path.abspath(os.path.dirname(__file__))
 class BaseConfig(object):
     """A class used to store coniguration properties common across
all environments"""
     DEBUG = True
     TESTING = False
     PORT = '9000'
```

a. **Cache** : In case you do enable caching, open the file `dbconfig.py` at location `test@testmachine:~/test/repodir/nyctaxiserver/app/configuration/` and set the value of the key variable **caching_enabled** to **True** (present in the dictionary **API_CONFIG**). You may enable caching for a particular API endpoints only (which should supports caching). For example in the below snapshot of file `dbconfig.py` caching is enabled for `total_trips` endpoint but not for `avg_speed24h` endpoint. Save and close `the file `dbconfig.py` after making changes.

```
 API_CONFIG = {

        'total_trips' : {
                         'bq_key':'bqconfig',
                         ....
                         'caching_enabled' : True,
                         ....
                        },

        'avg_speed24h' : {
                         'bq_key':'bqconfig',
                         ....
                         'caching_enabled' : False,
                         ...
                         },
             }
```

8. **Runing Tests** : In order to execute the testcases run the following command from shell. Before this please make sure to perform Step 2 (configuration) and Step 3 (Virtual Envirobment Set up) if not done already.

```
 -- Linux (verfied on Ubuntu 16.04.6 LTS):

     test@testmachine:~/test/repodir$ cd nyctaxiserver
```

```
     test@testmachine:~/test/repodir/nyctaxiserver$ source
venvtaxiapi/bin/activate/
     (venvtaxiapi) test@testmachine:~/testdir/repodir$ cd tests
     (venvtaxiapi) test@testmachine:~/testdir/repodir$ pytest
```

9. The execution trace logs are generated in the Folder below

```
 -- (venvtaxiapi)
test@testmachine:~/testdir/repodir/nyctaxiserver/app/main/Logs$
```