Principles of Urban Informatics

Homework #2

Instructor: Huy T. Vo – huy.vo@nyu.edu

Due: Mon 09/28/2015 9:30am

Overview and Preparation

For this assignment, you need to write a Python script to stream real-time bus data from MTA through the MTA Bus Time interface. In order to access this data, **you must first request an API key from MTA**. Here are the instructions:

1. Visit MTA Bus Time for Developers at http://bustime.mta.info/wiki/Developers/Index
2. Click on the "Go here" link to fill in your information and request an API. You should be receiving an email from MTA within an hour (most of the time within only a few minutes).
3. The key should be in the form of xxxxx-xxxxx-xxxxx-xxxxx-xxxxx
4. Please keep this key only to yourself as it would be your authorization token for MTA access and will be in this assignment.

Now, you're ready to access bus data from MTA! MTA is using the SIRI (Service Interface for Real Time Information) API to serve their data in both XML and JSON format. You are encouraged to use JSON for its increasing popularity in data access API over the web. In this assignment, we are only interested in tracking each vehicle journey (MTA provides both stop and vehicle information). More information on the vehicle monitoring stream is available at: http://bustime.mta.info/wiki/Developers/SIRIVehicleMonitoring

For example, using your key, you can retrieve all vehicle information for a bus line, e.g. B52, by accessing the following URL:
http://api.prod.obanyc.com/api/siri/vehicle-monitoring.json?key=**YOUR_KEY**&VehicleMonitoringDetailLevel=calls&LineRef=**B52**

IMPORTANT NOTE:

*The BusTime API server strictly enforces users to issue only 1 request per 30 seconds. Please do not constantly download or fetching data from the API in shorter periods. It is a good idea to **download a copy of the response to your local machine for testing purposes**. Also make sure to open it in an editor to familiarize yourself with all the fields of the response.*

Submission Info:

The solutions for Assignment 1 and Assignment 2 below must be pushed into your github repo under the named **PUI2015_<NetID>_hw2** (e.g. github.com/hvo/PUI2015_htv210_hw2).

CUSP
CENTER FOR URBAN SCIENCE+PROGRESS

NEW YORK UNIVERSITY

Assignment 1

You are required to write a Python script to retrieve and reporting information about active vehicle for a bus line. The final hand-in should be a single Python file, named *show_bus_locations.py* that takes exactly 2 arguments in the following format:

```
python show_bus_locations.py <MTA_KEY> <BUS_LINE>
```

where you can assume that <MTA_KEY> and <BUS_LINE> are valid and could be used to plug in the MTA URL above. For example, the program could be run as:

```
SAMPLE INPUT:
python show_bus_locations.py xxxx-xxxx-xxxx-xxxx-xxxx B52
```

The above command has to fetch data from the MTA website through the SIRI API using the provided key and return information on all available vehicles for the bus line B52. Your program will have to output the following to the console, which consists of the bus name, the number of vehicles and their current position:

```
SAMPLE OUTPUT:
Bus Line : B52
Number of Active Buses : 5
Bus 0 is at latitude 40.687241 and longitude -73.941661
Bus 1 is at latitude 40.690822 and longitude -73.920759
Bus 2 is at latitude 40.688363 and longitude -73.979563
Bus 3 is at latitude 40.688282 and longitude -73.979356
Bus 4 is at latitude 40.686839 and longitude -73.964694
```

Notes:

- Your program will be tested against the grader API key with different bus lines. Please make sure that your Python can read arguments from the command line.
- The bus line name is case-sensitive, aka. *B52* is not the same as *b52*
- All of your code must be put inside a single Python file named *show_bus_locations.py*
- You can assume that your Python will be run using the most recent Anaconda distribution. If you use any additional Python packages that are not included in Anaconda, please specify them in your README.md when you turn in your github repository.

CUSP
CENTER FOR URBAN SCIENCE+PROGRESS

NEW YORK UNIVERSITY

Assignment 2

In this assignment, we will build on the previous like to build on the previous assignment by adding more information to the buses. In particular, we would like display information on the next stop location of the bus. For example, whether the bus is approaching a stop, or is 1 stop away from it. Again, all these information are already included in the response JSON. We just need to extract them appropriately. Similar to the lab session, we will output the data to a CSV file instead of the decorated strings on screen. For buses that do not have this information, please output "N/A" in the stop information fields. The final hand-in should be a single Python file, named *get_bus_info.py* that takes exactly 3 arguments in the following format:

INPUT:

```
python get_bus_info.py xxxx-xxxx-xxxx-xxxx-xxxx M7 M7.csv
```

SAMPLE OUTPUT CONTENTS OF M7.CSV:

```
Latitude,Longitude,Stop Name,Stop Status
40.755489,-73.987347,7 AV/W 41 ST,at stop
40.775657,-73.982036,BROADWAY/W 69 ST,approaching
40.808332,-73.944979,MALCOLM X BL/W 127 ST,approaching
40.764998,-73.980416,N/A,N/A
40.804702,-73.947620,MALCOLM X BL/W 122 ST,< 1 stop away
40.776950,-73.981983,AMSTERDAM AV/W 72 ST,< 1 stop away
40.737650,-73.996626,AV OF THE AMERICAS/W 18 ST,< 1 stop away
```

Hints:

- The stop information are included in the "OnwardCalls" field nested somewhere in the JSON file.

Notes:

- All of your code must be put inside a single Python file named *get_bus_info.py*
- You can assume that your Python will be run using the most recent Anaconda distribution. If you use any additional Python packages that are not included in Anaconda, please specify them in your README.md when you turn in your github repository.
- When the OnwardCalls field is empty, you must output "N/A" as values for both the "Stop Name" and "Stop Status" fields.