Urban Computing Skills Lab

Module 3: Introduction to Python
Assignment #4

Instructor: Huy T. Vo - htv210@nyu.edu

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:

- Visit MTA Bus Time for Developers at http://bustime.mta.info/wiki/Developers/Index
- 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).
- The key should be in the form of xxxxx-xxxxx-xxxxx-xxxxx-xxxxx
- 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 in this assignment because it is more lightweight as well as more efficient to process. In addition, 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. B38, by accessing the following URL:
http://api.prod.obanyc.com/api/siri/vehicle-monitoring.json?key=**YOUR_KEY**&VehicleMonitoringDetailLevel=calls&LineRef=B38

You would get back a JSON string where all vehicles information are listed under this path (shown as a Python's dictionary):

data['Siri']['ServiceDelivery']['VehicleMonitoringDelivery'][0]['VehicleActivity']

Problem 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 *get_data.py*, that takes exactly 2 arguments in the following format:

```
python get_data.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 get_data.py xxxx-xxxx-xxxx-xxxx-xxxx B38
```

The above command has to fetch data from the MTA website through the SIRI API using the provided key and return the all available vehicles for the bus line B38. Your program will have to output the following to the console, which consists of the number of buses and their current position:

SAMPLE OUTPUT:

```
Bus Line               : B38
Number of Active Buses : 25
Locations:     Lat/Lon
(40.692547,-73.928113)
(40.700990,-73.904617)
(40.693593,-73.990473)
(40.696103,-73.991070)
(40.693972,-73.929599)
(40.699476,-73.923982)
(40.691680,-73.948708)
(40.691584,-73.987666)
(40.699400,-73.924058)
(40.708958,-73.912110)
(40.689833,-73.951728)
(40.697910,-73.925519)
(40.698289,-73.925147)
(40.712764,-73.915355)
(40.704733,-73.918798)
(40.687076,-73.975676)
(40.691100,-73.953750)
(40.694504,-73.990585)
(40.692447,-73.989064)
(40.691512,-73.987487)
(40.691183,-73.953025)
(40.689660,-73.966248)
(40.691408,-73.987225)
(40.690601,-73.945033)
(40.687919,-73.968324)
```

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.
- Suggested packages to be used in this script: json, sys and urllib2
- Documentation (including examples) of how to use Python's json package is available at: https://docs.python.org/2/library/json.html (checkout json.loads())

- urllib2 can be used to retrieve web contents given an URL. Examples and documentation are available here: https://docs.python.org/2/library/urllib2.html
- To retrieve program arguments, please look into sys.argv variable: https://docs.python.org/2/library/sys.html

Problem 2

All locations returned by MTA Bus Time are in geographic coordinates (latitude and longitude). While this is standard in most location services, for processing urban datasets in many GIS systems, different coordinate may be required. In Problem 2, you are tasked to convert the returned locations from MTA Bus Time to "NAD83 / UTM zone 18N" (ESRI:26918) coordinate, which is being used many shapefiles provided the New York State GIS Clearing House.

Coordinate conversion is always a tedious task, fortunately, the pyproj library provides a neat (and robus) API for us to take care of this. For example, the following Python codes would convert a geographic coordinate into ESRI:26918:

```
import pyproj
proj = pyproj.Proj(init="esri:26918")
nyc_lon = -73.928113
nyc_lat = 40.692547
print proj(nyc_lon, nyc_lat)
```

Please note that the *pyproj* package would need to be installed for the program to work. More information on how to install pyproject is available at https://pypi.python.org/pypi/pyproj/

The assignment is for you to add coordinate conversion to your get_data.py in the Problem 1 to print out the ESRI:26918 coordinate along with the geographic coordinates. Below are the sample input and output of the program:

INPUT:

```
python get_data.py xxxx-xxxx-xxxx-xxxx-xxxx M7
```

SAMPLE OUTPUT:

```
Bus Line               : M7
Number of Active Buses : 21
Locations:     Lat/Lon                ESRI:26918
(40.762663,-73.978405)   (586225.84,4512912.89)
(40.757224,-73.986097)   (585583.58,4512301.59)
(40.794290,-73.970194)   (586877.67,4516431.84)
(40.798969,-73.963068)   (587472.71,4516958.33)
(40.810089,-73.943893)   (589075.41,4518212.05)
(40.760947,-73.983369)   (585809.06,4512717.53)
(40.784504,-73.977336)   (586287.80,4515338.46)
```

```
(40.805026,-73.956809)   (587992.71,4517636.97)
(40.753436,-73.985127)   (585670.32,4511882.04)
(40.782972,-73.978454)   (586195.45,4515167.30)
(40.804520,-73.955608)   (588094.68,4517582.01)
(40.820287,-73.936263)   (589705.22,4519351.90)
(40.747517,-73.993153)   (585000.33,4511217.18)
(40.802179,-73.950051)   (588566.53,4517327.73)
(40.820955,-73.935853)   (589738.90,4519426.47)
(40.770740,-73.982246)   (585891.24,4513805.73)
(40.741346,-73.993935)   (584942.17,4510531.40)
(40.772318,-73.982087)   (585902.63,4513981.06)
(40.737845,-73.996482)   (584731.56,4510140.30)
(40.776746,-73.981991)   (585905.02,4514472.69)
(40.765766,-73.979858)   (586099.19,4513255.92)
```

Notes:

- Please turn in your new get_data.py as get_data_proj.py