labrinid Update README.md b97519b 7 minutes ago

1 contributor

163 lines (136 sloc) 7.66 KB

Repository: 2017-09.project-1.template

Assignment #1: Python

> Course: **CS 1656 - Introduction to Data Science** (CS 2056) -- Fall 2017
> Instructor: Alexandros Labrinidis
>
> Assignment: #1.
> Released: September 12, 2017
> **Due: September 25, 2017**

Description

This is the **first assignment** for the CS 1656 -- Introduction to Data Science (CS 2056) class, for the Fall 2017 semester.

Goal

The goal of this assignment is to familiarize you with the Python programming language and also expose you to real data and real APIs.

What to do -- wheresmybus.py

You are asked to write a Python program, called `wheresmybus.py` that will act as an interface to the TrueTime API v3 of the Port Authority of Allegheny County that provides real-time information on bus arrivals.

You can find more information about TrueTime at
http://www.portauthority.org/paac/CompanyInfoProjects/DeveloperResources.aspx

You program should be invoked as:

```
python3 wheresmybus.py command optional_arguments
```

There are four possible options for *command* and *optional_arguments*; these are specified next.

(1) wheresmybus.py getroutes

`wheresmybus.py getroutes` should connect to the TrueTime API and download all the available bus routes, using the `getroutes` call. Your program should do two things:

- print the route (rt) and route name (rtnm) fields separated by a comma, one route per line, e.g.,:

```
61D, MURRAY
65, SQUIRREL HILL
```

- save the data as a json object, in a file named `allroutes.json`, as follows:

```
[{'rt':'61D', 'rtnm':'MURRAY'},
 {'rt':'65',  'rtnm':'SQUIRREL HILL'}]
```

Please note that the above file does not contain all routes, it is merely illustrating the formatting.

(2) wheresmybus.py getdirections

`wheresmybus.py getdirections` should connect to the TrueTime API and download the available directions for all routes from `allroutes.json` (generated by the previous command) **that start from a `6`** (to keep things manageable). You should use the `getdirections` call and keep in mind the rate limitations for the TrueTime API (10,000 requests per day), so you should implement a delay timer. As before, your program should do two things:

- print the available rt/rtnm/direction combination for each route separated by a comma, one combination per line, e.g.:

```
61D, MURRAY, INBOUND
61D, MURRAY, OUTBOUND
65, SQUIRREL HILL, INBOUND
65, SQUIRREL HILL, OUTBOUND
```

- save the data as a json object, in a file named `6routes.json`, as follows:

```
[{'rt':'61D', 'rtnm':'MURRAY', 'dir':'INBOUND'},
 {'rt':'61D', 'rtnm':'MURRAY', 'dir':'OUTBOUND'},
 {'rt':'65', 'rtnm':'SQUIRREL HILL', 'dir':'INBOUND'},
 {'rt':'65', 'rtnm':'SQUIRREL HILL', 'dir':'OUTBOUND'}]
```

If file `allroutes.json` does not exist, your program should quit after printing the following error message:

```
ERROR: file allroutes.json does not exist
```

(3) wheresmybus.py getstops routeID direction

`wheresmybus.py getstops routeID direction` should connect to the TrueTime API and using the `6routes.json` file download all the bustops for the routeID/direction combination specified, using the `getstops` call. As before, your program should do two things:

- print the available stop ID (stpid) and stop name (stpnm) for each stop, separated by a comma, e.g.:

```
18161, 5TH AVE + DIAMOND ST
2644, 5th Ave opp #2358
```

- save the data as a json object, in a file named `mystops.json`, as follows:

```
{'rt':'61D',
 'dir':'INBOUND',
 'stops': [
     {'stpid':'18161', 'stpnm':'5TH AVE + DIAMOND ST'},
     {'stpid':'2644', 'stpnm':'5th Ave opp #2358'}
 ]
}
```

Note that the above are partial results.

If file `6routes.json` does not exist, your program should quit after printing the following error message:

```
ERROR: file 6routes.json does not exist
```

If file `6routes.json` exists, but the specific routeID/direction combination is not there, then your program should quit after printing the following error message:

```
ERROR: invalid route/direction combination: routeID/directon
```

(where routeID/directon are the command line arguments to `wheresmybus.py getstops`).

Every execution of `wheresmybus.py getstops routeID direction` overwrites previous versions of file `mystops.json` , unless the program exists with an error message, in which case the previous file should persist.

(4) wheresmybus.py getarrivals stopID

`wheresmybus.py getarrivals stopID` should connect to the TrueTime API and download all the available predicted arrival times for the specified bus stop, using the `getpredictions` call. Your program should combine the downloaded data with the `6routes.json` file and do two things:

- print the rt/rtnm/rtdir/stpid/stopnm/timstmp fields separated by a comma, one arrival per line, e.g.:

```
61C, MCKEESPORT-HOMESTEAD, OUTBOUND, 30, Forbes Ave past Bouquet St, 20170912 01:04
61D, MURRAY, OUTBOUND, 30, Forbes Ave past Bouquet St, 20170912 01:05
```

- save the data as a json object, in a file named `myarrivals.json` , as follows:

```
[{'rt':'61C', 'rtnm': 'MCKEESPORT-HOMESTEAD', 'rtdir':'OUTBOUND',
  'stpid':'30', 'stopnm':'Forbes Ave past Bouquet St', 'timstmp':'20170912 01:04'},
 {'rt':'61D', 'rtnm': 'MURRAY', 'rtdir':'OUTBOUND',
  'stpid':'30', 'stopnm':'Forbes Ave past Bouquet St', 'timstmp':'20170912 01:05'}}]
```

If file `6routes.json` does not exist, your program should quit after printing the following error message:

```
ERROR: file 6routes.json does not exist
```

If file `6routes.json` exists, but the specific routeID is not there, then your program should simply use the value `TBD` for field `rtnm` .

Every execution of `wheresmybus.py getarrivals stopID` overwrites previous versions of `myarrivals.json` , unless the program exists with an error message, in which case the previous file should persist.

Important notes about grading

It is absolutely imperative that your python program:

- runs without any syntax or other errors (using Python 3)
- strictly adheres to the format specifications for input and output, as explained above.

Failure in any of the above will result in **severe** point loss.

Allowed Python Libraries (Updated)

You are allowed to use the following Python libraries (although a fraction of these will actually be needed):

```
argparse
collections
csv
json
glob
math
os
pandas
re
requests
string
sys
time
xml
```

If you would like to use any other libraries, you must ask permission by Friday, September 15, 2017, using piazza.

About your github account

It is very important that:

- Your github account can do **private** repositories. If this is not already enabled, you can do it by visiting https://education.github.com/
- You use the same github account for the duration of the course
- You use the github account that you specified during the test assignment (i.e., this one)

How to submit your assignment

For this assignment, you must use the repository that was created for you after visiting the classroom link. You need to update the repository to include file `wheresmybus.py` as described above, and other files that are needed for running your program. You need to make sure to commit your code to the repository provided. We will clone all repositories shortly after midnight:

- the day of the deadline **Monday, September 25th, 2017 (i.e., at 12:15am, Tuesday, September 26, 2017)**
- 24 hours later (for submissions that are one day late / -5 points), and
- 48 hours after the first deadline (for submissions that are two days late / -15 points).

Our assumption is that everybody will submit on the first deadline. If you want us to consider a late submission, you need to email us at `cs1656-staff@cs.pitt.edu`