GrubHub.com Scraping by HaoHsin Shih

LinkedIn: http://www.linkedin.com/in/haohsinshih • Github: https://github.com/BryantShih

1. Testing environment and necessary libraries (make sure you have these libraries before running my code):
   - I developed this project by python 2.7 and terminal tool on system OS X Yosemite.
   - Python library – selenium: (For manipulating chrome driver)
   - Chrome driver: (For solving some tricky challenge by simulating chrome browser's behavior, so Google Chrome is needed as well. Download the version for you system and put it in the folder of my project. There should be one in my project already which is for Mac32, substitute it with correct one to your system)
     http://chromedriver.storage.googleapis.com/index.html?path=2.20/
   - Python library – xlrd: (For reading excel file)
     ➢ https://pypi.python.org/pypi/xlrd
   - Python library – xlwt: (for writing excel file)
     https://pypi.python.org/pypi/xlwt
   - Python library – dryscrape: (For getting javascript-drived contetnt on web page)
     http://dryscrape.readthedocs.org/en/latest/installation.html
   - lxml: (For parsing option to Beautiful soup)
     http://lxml.de/installation.html
   - Python library – Beautiful soup: (For parsing and formatting html tag)
     http://www.crummy.com/software/BeautifulSoup/#Download
   - Ot, Qt Webkit : (Necessary library for dryscrape)
     http://www.qt.io/download/

2. How to run my project:
   After all the libraries are set, the rest for running my code is very simple.
   Just run terminal tool and switch to the folder of my project then run:

   ```
   python ScrapeGrubHubByCity.py "city_name"
   ```

   city_name could be New York, Philadelphia, Houston, Denver, Sacramento, Portland, Hartford, Las Vegas.

   Then the program will do the scraping automatically. During scraping, avoid using you browser to operate GrubHub.com because it could change your cookie setting and affect our scraping.

   The file "HARTFORD_restaurant_data.xls" in my project is a sample output for city Hartford.

3. Future work:
   This scraping program would spend ton of time on loading web page content and

checking if the content is as complete as one we watch on browser. So actually most time for running this program is about web pages loading but the work of parsing and scraping itself is very efficient by the integration of existing libraries.

The efficiency for my codes could be optimized by making it a multi-thread program(it's currently a single-thread worker) and adding different proxy setting to avoid the loading limit. There are still lots of challenges to polish this project but I think it could satisfy the basic requirement so far.