

Lab on Urban Data Handling

09/16/2015



The City College
of New York

OF THE CITY OF NEW YORK

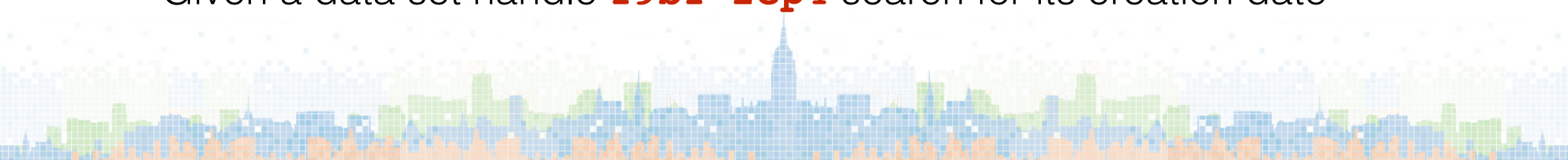
Objectives

- Familiarization with online data portals and data representation
- Hands-on experiences with static feed, URL request and JSON manipulation in Python
- Setting data services at NYU: CartoDB and NYU Box



Task 1: access metadata from Web UI

- (Re-)Introducing NYC OpenData : <https://nycopendata.socrata.com>
 - A comprehensive data dumps (more than needed) of open data in NYC
 - Each data has a unique handle
 - Metadata is included with each data
 - Data can be exported into CSV, XLS, etc.
- Given a data set handle **f9bf-2cp4** search for its creation date



Task 2: access metadata through API

- Find out the creation date of another data set **h9gi-nx95**
 - Tedious task to search, then several more clicks
- Let's do this through the API
 - metadata are available at:
<https://nycopendata.socrata.com/views/h9gi-nx95>
 - and are in JSON format! (enter the URL into your browser and notice the *createdAt* field)



Task 2 — continued

- Fetch the metadata to a file:

```
curl https://nycopendata.socrata.com/views/h9gi-nx95 > metadata.json
```

- Write a Python script, *task2.py*, to output the *createdAt* time as a human-readable string:

```
bash$ python task2.py metadata.json  
2014-04-28 12:41:44
```



Task 3

- Instead of using curl to download the metadata first, add to the Python script the ability to download this data given a data handle as well

```
bash$ python task3.py h9gi-nx95  
2014-04-28 12:41:44
```

```
bash$ python task3.py f9bf-2cp4  
2013-02-20 22:28:53
```



Task 4

- Find the list of CitiBike stations that are “coming soon” (any station that doesn’t have the status key of 3 and its name starts with “Coming soon”:

http://www.citibikenyc.com/stations/status_json

- Fetch the current feed of the stations and save to stations.json:

- <https://www.citibikenyc.com/stations/json>

- Write a Python script to list list all the station names and locations:

```
bash$ python task4.py stations.json
S 4 St & Rodney St : 40.70934,-73.95608
...
```



Task 5

- Make the script in Task 4 output to a CSV file instead

```
bash$ python task4.py stations.json comingsoon.csv
```

- And load `comingsoon.csv` onto CartoDB (setup CartoDB instruction)
- Make sure to output a proper header for CartoDB to auto-recognize the geo-referencing columns

