Final Project: Web Service APIs

This final project should serve to bring together almost all of the topics we have been covering throughout this class, from manipulating tidy data to working with hierarchical data to the underpinnings of client-server computing and working with Provider APIs.

It is also intended to be self-defined. Do some exploration and find data from a provider that you are interested in investigating.

The final products of this project are two-fold:

1. You will produce and turn in a Notebook that gives documented processing code, organized as a set of functions, that shows your Python programming steps of data acquisition, traversing data and building normalized (tidy) data tables, and the export of the tidy data as CSV files to be used to enable visualizations. It should also include code for some preliminary data exploration, like histograms or scatter plots used to better understand the data.

   > This is equivalent to the guided notebooks I prepared for you in the movie project, both the processing and the basic exploration. Now, you are expected to design, write, structure, and debug the code for yourself.

   > You will also use this notebook to **explain** your design and processing decisions and experiences.

2. You will produce an essay in which you develop a **data story**. A data story is a single, compact, thesis that drives your work from start (inquiry) to end (conclusion). Use the data exploration phase during your notebook processing to find a compelling storyline in the data you can utilize for your project. Like in the mid-semester project, your deliverable is a **PDF** with your essay. This could be written in markdown, in LaTeX, in Google docs, or in Word. The basic assumption is that the data story will develop at least three interesting questions that "rise above" obvious exploration.

The essay will include figures generated through some visualization tool, such as Tableau, but other document composition tools are possible. Regardless of tool, you are to turn in a **PDF** of the resultant essay.

Requirements

The requirements of this project include the following:

1. Work with a **Web API** designed by a Data System provider.
   - The selected provider must require authentication, but this authentication may either be by API key or through the more involved `OAuth`.
   - The API must use HTTP for its requests and responses.
   - See https://github.com/public-apis/public-apis for a fairly comprehensive table listing of public and semi-public APIs along with the type(s) of authentication used by the API.
   - Some other online listings of APIs:
     - https://rapidapi.com/blog/most-popular-api/
     - https://any-api.com/
2. Find a second data source, which could be openly provided, or could be obtained through web scraping techniques, that **complements the data** from the first data source.
   - The second data source could also be API-based.
   - The second data source **cannot** be static links to CSV/XML/JSON data files.

3. Use the skills/knowledge from the Hierarchical Data Models unit to parse and extract tables of information from XML, JSON, and HTML from your providers. You **must** end with **tidy** data tables (with explicitly documented functional dependency), and have these tidy tables in `pandas`.
4. Practice good software development techniques:
    - Practice functional abstraction and associated code documentation
    - Program defensively, checking for error codes/returns in *every* client/server interaction, and handling the error in a reasonable way
5. Give the acquired data meaning through your data story and essay to ask and present findings for at least three interesting questions about the data.

Variations

The expectation is that students work in teams of two, collaborating and working together (as opposed to "you do this half and I'll do another half"). Teams will likely be self-selected but, like for quiz study groups, I will provide a form for students to "opt-in" for professor-generated random pairing.

Single-person and triple person teams are possible, but the expectation will be different than the two-person team expectation:

- A one person team is allowed to use a single authenticated data source, but they must develop and use multiple endpoints supported by the provider. A one person team may also reduce the number of interesting questions from three to two.
- A three person team is expected to do work commensurate with 50% more than the work of a two-person team:
    - On the data source side, this could mean 3 data sources, or could mean use of the more sophisticated OAuth2 security and use of more endpoints.
    - The team could design and populate a **relational database** with their resultant tidy data (instead of growing to additional data sources).
    - The team should have 4 to 5 interesting questions as part of their data story, depending on the aggregate level of sophistication of the questions.

Process

Projects will be due the last day of regular exams by **5pm, May 18th**. Students are **strongly encouraged** to start early.

Projects that are completed by **5pm May 16** will be given 5% extra credit.

The project specifics are left to the creativity and design by the student involved, but must satisfy all of the above goals. Explore the Providers and APIs, perhaps coming up with providers that are not included in the provided lists, and think about data that excites/intrigues you.

Some OAuth Data Providers

1. Facebook
    - **Graph API** allows querying and posting from a program and includes ability to get lists of friends and many other
2. LinkedIn
    - Uses OAuth 2
    - RESTful APIs
3. Fitbit
4. Tumblr

5. GitHub

6. Twitter

    - Note that this uses OAuth 1, not 2, so there will be additional learning curve
    - There are streaming APIs, but these will require additional complexity on the programming side, and to be useful, you will want to start early so that you have sufficient time to collect data on a stream to allow for interesting analysis

7. Dropbox

    - Data is relative to whatever one deposits into Dropbox, so this will need something additional to generate interesting data.

8. Google Drive

    - Same comment as for `Dropbox`