

# Web APIs and Persistent Storage

## DS3500: Advanced Programming with Data

### Purpose

- Practice using the *requests* library to call web-based APIs
- Store response data in a database such as Mongo (recommended), SQLite, or MySQL.
- Explore a data source of your own choosing and techniques for data persistence
- Implement a mini-mini project in 1 week

### Overview

In this assignment you will persist data obtained through a web API to a database. We recommend you use Mongo, a document store that stores data in JSON format. This will be especially convenient for web-based APIs that respond with JSON-formatted data. If you prefer to use a traditional relational database such as SQLite or MySQL, you will need to implement your normalized schema and do all the necessary object-relational mapping. Mongo is the most popular NoSQL (non-relational) database on the market today.

**Step 1:** Choose a data source with a web-based API. You can pick any data source you like but you must use the native *requests* library to retrieve data from your data source. You are not allowed to use a python wrapper that might be available for your APIs. Here is an extensive list of web-api-enabled data sources: <https://github.com/realpython/list-of-python-api-wrappers>. Listed are the associated python wrappers that you must avoid using for purposes of this assignment. We want you to demonstrate a solid understanding of how to use the native *requests* library.

**Step 2:** Programmatically retrieve data from the source using the requests library and store the data in your database. Your response should have at least 200 datapoints.

**Step 3:** Ask a well-defined question about your data. Try to ask a question that would reveal something interesting and non-obvious about your data.

**Step 4:** Translate your question into 2-3 database queries. These queries should be submitted to the database programmatically (but of course you can test them manually using a terminal console or the MySQL workbench if you are using MySQL or Compass if you are using Mongo.)

**Step 5:** Visualize your query result and explain and declare your conclusions. We are looking for a creative visualization. Give us more than a line plot or a bar chart. A network/graph visualization using the *networkx* library or an interactive visualization using *plot.ly* would be very nice. I may share the best results with the class.

*Pair programming in groups of 2 (maximum) is allowed for this assignment.*

### What to submit

Submit your python code (.py) and a one-page summary report (.pdf) that documents your data source, the data that was retrieved, the question you asked of your data, your resulting visualization, and any conclusions.