Data Gathering, Web Automation & GIS

Wael Elhaddad
NHERI SimCenter
Programming Bootcamp 2019 (Day 4)
Outline (Day 4)

- **Introduction**
  - Web Technologies & HTTP
  - Web APIs (e.g. REST)
  - JSON
  - Relevant Web Services (Exposure and Hazard Data)

- **Web Automation using Selenium**
  - Tax Assessor’s Data (e.g. Anchorage, Memphis, NJ...etc.)

- **Visualization & Analysis in GIS**
  - Introduction to QGIS

- **AI Applications**
  - Computer Vision
  - Data Enhancement (SURF)

- **Regional Data Gathering Exercise**
Web Technologies

What happens when you open the browser and type www.google.com?
**Hypertext Transfer Protocol (HTTP)**

What happens when you open the browser and type [www.google.com](http://www.google.com)?
Then, what happens when you search for something?

![Diagram of HTTP request and response](image)

**Request**

**Response**

(e.g. HTML, XML, JSON...etc.)
Web API

- **Application Programming Interface (API)**
  - Defines a set of methods for communication

- **Web API**
  - Defines the methods for communication between a client and a server

- **REST API**
  - Set some standard rules for web communication (e.g. HTTP)
  - Four methods are defined (GET, POST, PUT, DELETE)
    - GET: to retrieve data
    - POST: to create data
    - PUT: to modify data
    - DELETE: to delete data
JavaScript Object Notation

File format to describe data in human-readable form

The format provides attribute-value pairs

Data Types
- Number
- String
- Boolean
- Array
- Objects

Disadvantage: large size (not efficient)
Web Services

▪ ATC API
  ▪ Hazard by Location API: https://hazards.atcouncil.org/api
  ▪ Example: https://api-hazards.atcouncil.org/wind.json?lat=35.4676&lng=-97.5164

▪ USGS APIs (NSHMP-ws)
  ▪ Hazard Service: https://earthquake.usgs.gov/nshmp-haz-ws/
  ▪ Design Maps: https://earthquake.usgs.gov/ws/designmaps/

▪ FDNS
  ▪ Earthquake Catalog: https://earthquake.usgs.gov/fdsnws/event/1/
  ▪ Examples:
    - Ridgecrest, CA
    - Anchorage, AK
Web Services

- **DataSF Portal**
  - Tall Building Inventory
    - Inventory: [https://data.sfgov.org/Housing-and-Buildings/Tall-Building-Inventory/5kya-mfst](https://data.sfgov.org/Housing-and-Buildings/Tall-Building-Inventory/5kya-mfst)
    - Request: [https://data.sfgov.org/resource/5kya-mfst.json](https://data.sfgov.org/resource/5kya-mfst.json)

- **Census API**
  - [https://www.census.gov/data/developers/data-sets.html](https://www.census.gov/data/developers/data-sets.html)
Python Libraries

- **Requests**
  - Submit HTTP requests and get the response

- **Selenium**
  - Webdriver to control the web browser
  - Documentation: [https://selenium-python.readthedocs.io/getting-started.html](https://selenium-python.readthedocs.io/getting-started.html)

- **BeautifulSoup, lxml**
  - Packages to facilitate processing html

- **Census, US**
  - Python package to facilitate querying Census data
  - Documentation: [https://github.com/datamade/census](https://github.com/datamade/census)
Requests Demo

- Using requests we will get a list of tall buildings and print one of them to the screen

```python
import requests

# Let's request the tall buildings information
response = requests.get("https://data.sfgov.org/resource/5kya-mfst.json")

# Let's check the response
if(response.status_code == 200):
    tallBuildings = response.json()

    print("Building Name", tallBuildings[0]["name"])
    print("\tOccupancy: ", tallBuildings[0]["occupancy"])
    print("\tAddress: ", tallBuildings[0]["address"])
```

- **Exercise 1**
  Print to the screen the list of buildings including relevant information about the building like structure type, occupancy, number of stories, total area.

- **Exercise 2**
  Write the data from exercise 1 into a csv text file, including the latitude and longitude.

- **Exercise 3**
  Can we get PGA from USGS API for each building and include it in the output file?
Using Selenium, we automate browsing the tax assessor’s website

Exercise 4: Can we extract more information about these buildings e.g. number of stories, year built, area...etc.

Exercise 5: Let's do the same for Memphis, Tennesse

http://www.muni.org/pw/gsweb
GIS Introduction

- GIS stands for Geographical Information System
- Information is represented in a set of layers
- GIS platforms can help you:
  - Generate maps & visualize geospatial data
  - Transform and edit data
  - Perform spatial analysis on the data (e.g. spatial joins)
GIS Software

- **ArcGIS (Commercial)**
  - Desktop & Online (cloud/web-based)
  - Many universities provide access to student, staff and faculty

- **QGIS (Free & Open-Source)**
  - Desktop only
  - Easy to use
  - Extensible using Python
GIS Basics

- Coordinate Systems (CRS)
  - Map Projection
  - There are many systems (e.g. Local CRS)
  - Latitude and Longitude (WGS84 EPSG:4326)
GIS Basics

- Two Types of Data Layers
  - **Vector Data**
    Suitable for discrete and distinct features
    e.g. Buildings, Roads...etc
  - **Raster Data**
    Suitable for continuous features
    e.g. elevation, temperature, soil properties...etc
GIS Basics

- **Vector Data**: Geometry and Attributes

  **POINTS**: Individual \(x, y\) locations.
  - Examples: Center point of plot locations, tower locations, sampling locations.

  **LINES**: Composed of many (at least 2) vertices, or points, that are connected.
  - Examples: Roads and streams.

  **POLYgons**: 3 or more vertices that are connected and **closed**.
  - Examples: Building boundaries and lakes.