Scraping Postal Codes of all Airports in the world (blueprint)

Subhankar Ranjan Paul
February 27, 2020

1 Problem Statement

Find postal codes of all airports in the world

2 Intuition

If we can find the web pages where from we can get the postal codes of airports, we can crawl through those web pages and scrap all the postal codes(IATA , ICAO) along with the airport names.

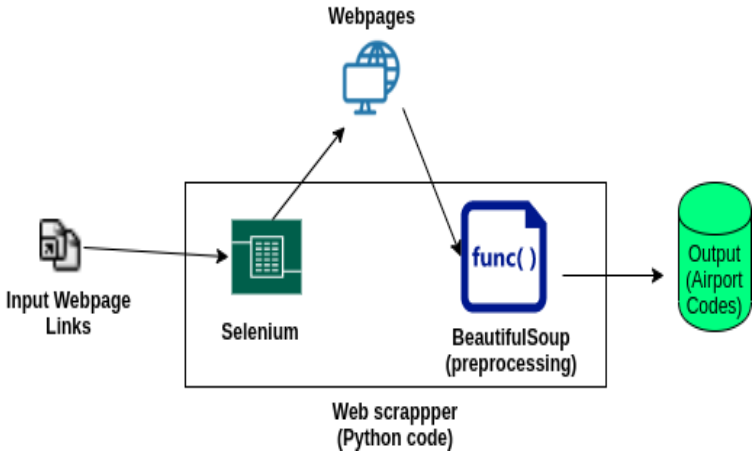
3 Technology Stack

- Language: Python
- Libraries:Selenium,BeautifulSoup,Pandas,CSV
- Browser:Firefox
- Web Driver: Geckodriver to aid selenium automation.

4 Looking for web pages with airport codes

As the first step towards accomplishing this task we have to surf the web to find web pages where we can find postal codes. Web pages likeworld-airport-codes, nationsonline host such information.But the wikipedia page List_of_airports_by_IATA_code:_A is found to be most optimal for extracting the postal codes. As in here, the airport codes are stored in tabular format. Specifically, IATA code is in the first column, ICAO the second and AIRPORT NAMES in the third of the table.

5 Implementation



System Architecture

Above figure represents the system architecture in which the web page links were fed as input to selenium, which crawls through them and handovers each page source to BeautifulSoup which further extracts the postal codes.

5.1 Inspecting Page Source

Inspecting the page source of wikipedia ***List_of_airports_by_IATA_code:_A*** it was found that there are 26 unique pages for postal codes starting with each english alphabet. Also it was noticed that *the url for each page varies only at the last character.* For example for airport names starting with A the url is ***en.wikipedia.org/wiki/List_of_airports_by_IATA_code:_A*** and for airport names starting with B the url is ***en.wikipedia.org/wiki/List_of_airports_by_IATA_code:_B***

We can exploit this property to crawl through different web pages extracting the airport codes starting with different alphabets. The postal codes and corresponding names are stored in table with class name ***wikitable sortable***.

5.2 Firefox automation using Selenium

Web scrapping with BeautifulSoup alone overlooks JavaScript thus information rendered dynamically can't be extracted using BeautifulSoup alone. For this, Selenium comes into the rescue by automating browser interaction from Python. Hence the information rendered by JavaScript links are clicked automatically through Selenium which can later be handed over to BeautifulSoup for further extraction. By using Selenium WebDriver we can crawl through the ***wikipedia*** web pages for each alphabet automatically through Firefox, and handing over the page source to BeautifulSoup.

5.3 Extracting Postal Codes using BeautifulSoup

The page source received is extracted using BeautifulSoup by iterating through every row of the table with class name ***wikitable sortable***. Iterating through every row once a *'table'* is found and adding the first column(IACA code) ,the second(ICAO code) and the third(Airport NAME),in the list ***postal_codes***.

5.4 Storing Postal Codes in Pandas Data Frame

Pandas data frame can be used to store the extracted information in a structured manner, also it makes the visualisation of data fascinating.

5.5 Storing in a seperate file(CSV format)

It is often preferable to store the extracted information in a desired format.In our case, we can prefer to store data in a seperate file named ***'Airport_Code.csv'*** in a Comma Seperated Value (CSV) format, where in the IATA,ICAO and AIRPORT NAMES are represented in first, second and third column respectively .

6 Conclusion

Through this task the postal codes of airports were successfully extracted and in depth knowledge of **web crawler, web scrapper , Selenium, BeautifulSoup, Pandas, CSV** is gained. As a future scope various NLP techniques can be deployed to obtain better structured data and to further enhance the input pipeline process such as preprocessing the input data. Later we can also also extend it further with concepts like big data (Hadoop File System) to deal with large amount of data sets.

This project further intensified my appetite in the data engineering domain.