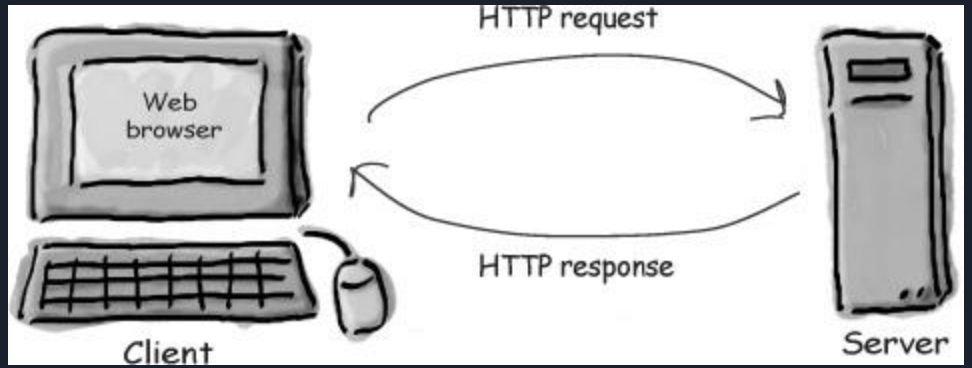
Programmer Guide to webscraping

Webscraping using python and selenium
By:Saikrishna

Prerequisite

- Familiarity of Python language
- Knowledge of Html and CSS
- Understanding of Http verbs like Get and Put

Http- The what and How?



What is Web Scraping?

Web Scraping (also termed Screen Scraping, Web Data Extraction, Web Harvesting etc.) is a technique employed to extract large amounts of data from websites whereby the data is extracted and saved to a local file in your computer or to a database in table (spreadsheet) format.

Why you should scrape?

- Api may no be provided what you need
- No need to worry about rate limits
- Extract data what you need.
- Reduce manual efforts.

Things that help in webscraping

- Python Libraries(Urllib, Beautifulsoup, Scrapy)
- Xpath
- Regular expressions
- Selenium

How is it done

It is broadly classified into 3 steps

- ➔ Getting the Html content
- ➔ Parsing the response data
- ➔ Optimising and preserving the data in excel or database

Getting the Content

- Getting the Url
- Using Http libraries to make request
- Fire GET/POST request to the server
- Capture the response which need to extracted

Extracting the data

1. Using Basic python along with Regular Expression
2. Using python libraries to parse the content

   a) Using Beautiful Soup and Lxml

   b) Each modules have its own technique.

Beautiful Soup

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

LXML

The lxml XML toolkit is a Pythonic binding for the C libraries libxml2 and libxslt. It is unique in that it combines the speed and XML feature completeness of these libraries with the simplicity of a native Python API, mostly compatible but superior to the well-known ElementTree API.

Preserving the Data

- ➔ Writing to a file
- ➔ Exporting data into CSV or Excel
- ➔ Storing to the Database.

Example Use Case

An use case to get the list of popular hotels for all states

and build   datamodel.

Lets see the Code!

Code Sample

```
options = webdriver.ChromeOptions()

options.add_argument('headless')

driver =
webdriver.Chrome(chrome_options=options)

page=driver.get(url)

soup = BeautifulSoup(page.content,
'html.parser')
```

Too Many ways of Scraping?

Use Cases about the usage

- Dynamically generated HTML pages->Use Selenium
- Cookie based scenarios → Use either of request module/ Selenium

Ethics of Scraping

- Read privacy policies before scraping
- Do not Brute force or perform DDOS attacks
- Publishing scraped content may be a breach of copyright.
- Conform robots.txt file.

QUESTIONS?

THANK YOU