International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 9 Sep. 2022, pp: 235-237 www.ijaem.net ISSN: 2395-5252

Web Scraping Using Python

Sanit Kumar, Jyoti Thakur, Deepmala Ekka, Ishwar Sahu
B.Tech 3rd Year Student, Computer Science and Engineering, Government Engineering College Bilaspur, Chhattisgarh, India

Date of Submission: 01-09-2022 Date of Acceptance: 10-09-2022

ABSTRACT –
The internet is a huge collection of heterogeneous data. It is the origin of vast information and data source from

where we can get information about anything that present in the world. However, we also know that every information or data on the internet is in unstructured or poorly structured form. Therefore, it takes more time to search the data according to the needs of the users because not every information on the internet is useful to us. Therefore, it becomes a challenge for us how we will get useful data in less time. One of the solutions for this kind of problem is web scraping. This article offers an introduction about static and dynamic web scraping for useful data extraction, and it provides a basic idea about extracting, storing, and reusing data. With the help of

web scraping, we can convert the unstructured data into a structured form.

Key Words: Virtual Environment, BeautifulSoup, Requests, Selenium, ChromeDriver, Pandas, Keys, CSV, XML.

I. INTRODUCTION

When we talk about web scraping the first thing that comes to our mind is the web pages because this page can contain a large amount of data. The process of extracting something out of web pages through a web scraper by using HTML or a web browser is known as web scraping. We may store this scraped information in a database and later we can use this data for another purpose somewhere else. Nowadays we find many websites on the internet, which is used for different purposes, and we know that a website is a collection of related web pages that may contain text images audio and videos. A website can consist of one or thousands of pages depending on what the site owner is trying to do. There are two types of web pages - static web pages and dynamic web pages. in the scraping process, we will find that the scraping of static pages is much easier than the scraping of dynamic pages because in static scraping we simply send an HTTP request to the server and in response from the server we will get the exact page we are looking for without doing any other additional process. On the other hand, the scraping of dynamic pages is a very complex task because almost every dynamic page is based on the server- side scripting languages like JavaScript, ASP, JSP and PHP. Therefore, it takes more time to get a load and it is possible that instead of an HTTP response we will get a JavaScript code in response from the server after sending an HTTP request. Hence, we can say in normal web scraping, we send an HTTP request to the website and get the page source and then we will convert it into soup object and find the elements. Now, if the site is JavaScript rendered, that is JavaScript provides the elements on the website, then sending an HTTP request will not fetch us the data. Therefore, in Dynamic Web Scraping, we will first let the website get loaded fully, then we will get the page source, and hence all the data is loaded first and then it is scripted. It also allows us to extract the dynamic or changing data from the web pages.
In this project, we used four Python libraries:

1.1 BeautifulSoup

This library is used to parse the html documents and defines the html tags. It creates a parse tree for parsed pages, which extract data through html. It receiving data from html, xml and other markup languages. Suppose you have found some web pages that display data relevant to your research, such as date or address information but this does not provide a way to download the data directly. Beautiful soup helps you pull special context from a webpage, remove html tags and save information.

1.2 requests

This is the most basic python library. It is used for http requests. This allows us to send http requests such as GET, POST etc. in websites so that we can retrieve data from the online websites. This library is simple and it can be used easily.

DOI: 10.35629/5252-0409235237

1.3 Selenium

It is an automation-testing tool for automating web browsers. It is an open source suite for different web browsers and platforms. Selenium supports different languages like c, java, python etc. It runs automation script smoothly. Selenium provides python binding APIs, allowing it to access web drivers such as chrome, Firefox etc. It provides flexibility and is free of cost.

It performs actions like:

□ Element clicking,
□ Page refreshing,
□ Go to website link.

1.1 Pandas

Pandas is a python library or a python package that provides an expressive data structure with high performance. It is a fast, flexible, and most powerful library widely used for data analysis, data manipulation, and data storing. It is an open- source for data analysis and data manipulation. Pandas provide a heterogeneous data type in a data frame. Pandas data frame contains two-dimensional data structure with row and column axis and by default, its create indexing begins with 0. It stores the data in CSV files, JSON, and Excel Sheets.

II. SETUP AND WORKING

2. SETUP AND WORKING

2.1 PLATFORM USED

PyCharm: The Integrated Development Environment used to write, compile and execute code specifically in Python Language.

2.2 INITIAL SETUP

2.2.1. Creating Virtual Environment

Creation of a separate Virtual Environment for the project is essential as it allows the program to install and use the Python libraries without creating any conflicts with other programs using the same libraries. It is required to create the Virtual Environment in the same folder as of the entire Project.

2.2.2. Installing the Python libraries

The Python libraries may be installed either by using the Command Prompt or by using the PyCharm IDE Terminal. We use the "pip" command for the installation of required libraries.

2.2.3. Programming the Web Scraper

We use the PyCharm IDE to create a Python program inside the Project folder. In the program, we import the installed python libraries first. The "webdriver" interface of Selenium library is used to implement the browser automation code. Request library code makes request to the web servers and BeautifulSoup library code is used to identify the HTML tags and parse the data from the web server. Pandas library code is used to convert the parsed data into desirable format.

III. WORKING MECHANISM

The Python program contains a function, which takes a word as an argument. The URL of a certain website is used to navigate to the desired web page by using Selenium "webdriver" interface. We use the "headless" option in the webdriver to hide the browser automation process. The search bar of the web page serves as a navigation tool for changing the web page contents. It can be used by putting the HTML tag for the search bar in selenium code. We can send the desired search elements by using the "Keys" module in Selenium. The word taken as the argument is then sent through Keys to the search bar. This word makes the website navigate to the desired web page. The current address of web page is taken as the source and all the required contents are then extracted using BeautifulSoup.

BeautifulSoup uses tags to identify the specific elements required for extraction. A Python dictionary element can be used to facilitate the proper storage and access of data elements. The dictionary element allows the data to be accessed in a structured manner using a key associated to the data element. During the execution of the Python program, an input is passed to the Scraper function that takes it as an argument and the data from the web page is extracted after function execution. We can now convert the extracted data into desired format using Pandas library.

After the program execution, a data file is created in the previously determined format that can be accessed to view the stored information.

3.1 Entity Relationship Diagram

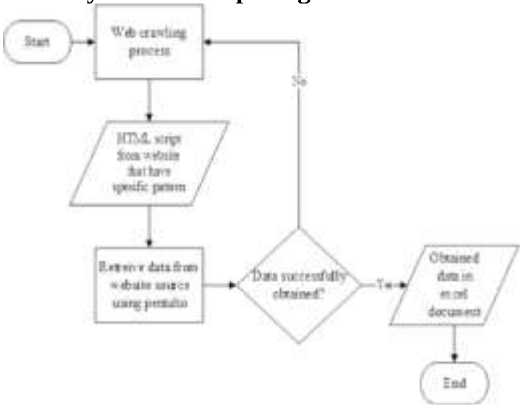
Start
Web crawling process
No
HTML script from website that have specific pattern
Retrieve data from website source using pentaho
Data successfully obtained?
Yes
Obtained data in excel document
End

Fig :- Entity-Relationship Diagram

IV. CONCLUSION

We have successfully created a Web Scraping which can extract the ever changing or dynamic data from the web pages. The Dynamic attribute of web page contents is achieved by the Web Scraper using the “search bar” of websites which allows us to change the contents of the web page by sending keywords through Selenium browser control. The data extracted from the website can be displayed in an understandable format.

ACKNOWLEDGEMENT

First, we would like to express our deep gratitude to our project guide Prof. Sanchita Chourawar (Department of Computer Science &amp; Engineering, GEC Bilaspur) for her patient guidance, enthusiastic encouragement and useful suggestions for this research work. We would also like to thank our respected Principal Dr. B.S. Chawla and Prof. Sourabh Yadav (Head of Department of Computer Science &amp; Engineering) of GEC Bilaspur for their assistance, advice and facilitating our project work.

REFERENCES

[1]. Anand V. Saurkar, KedarG. Pathare, and Shweta A. Gode, “An Overview OnWeb Scraping And Tools”, International Journal on Future Revolution in Computer Science &Communication Engineering (IJFRCSCE), Volume 4 Issue 4 ISSN:2454-4248, April2018

[2]. EMIL PERSON," Evaluating tools and techniques for web scraping", KTH ROYAL INSTITUTE OF TECHNOLOGY SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE, 2019

[3]. Katharine Jarmul and Richard Lawson, "python web scraping", May 2017

[4]. Deepak Kumar Mahto and Lisha Singh, "A dive intoWeb Scraper world", Institute of Electrical andElectronics Engineers (IEEE), 2016

[5]. David Mathew Thomas, and Sandeep Mathur, "Data analysis by web scraping using python", International Conference on Electronics, Communication and Aerospace Technology (ICECA), 450-454, 2019

[6]. Bar-llan, J. (2011). Data collection methods on the web for informetric purpose - A review and analysis. Scientometrics.

[7]. Markus Herrmann, and Laura Hoyden, "Applied WebScraping in Market Research", International Conference on Advanced Research Methods and Analytics, July 2016

BIOGRAPHIES

[1]. Mr. Sanit Kumar pursuing Bachelor of Technology with major in Computer Science and Engineering from Government EngineeringCollegeBilaspur(C.G.) Affiliated to Chhattisgarh Swami Vivekananda Technical University, Bhilai (C.G.) in 2022.Good WithPython, C, C++, HTML, Java programming, and Android, etc.

[2]. Miss. Jyoti Thakurpursuing Bachelor of Technology with major in Computer Science and Engineering from Government EngineeringCollegeBilaspur(C.G.) Affiliated to Chhattisgarh Swami Vivekananda Technical University, Bhilai (C.G) in 2022.Good with Python, Django, and DataScience.

[3]. Miss. Deepmala Ekkapursuing Bachelor of Technology with major in Computer Science and Engineering from Government Engineering College Bilaspur (C.G) Affiliated to Chhattisgarh Swami Vivekananda Technical University, Bhilai (C.G) in 2022. Good with Python, C,andC++ programming

[4]. Mr. IshwarSahupursuing Bachelor of Technology with major in ComputerScienceandEngineering from Government Engineering College Bilaspur (C.G) Affiliated to Chhattisgarh Swami Vivekananda Technical University, Bhilai (C.G) in 2022. Good with Python, C,andC++ programming