A Dive into Web Scraper World

Deepak Kumar Mahto
HMR Institute of Technology and Management
Delhi, INDIA
Email ID: deepak7mahto@gmail.com

Lisha Singh
HMR Institute of Technology and Management,
Delhi, INDIA
Email ID: lisha.engg@gmail.com

Abstract* – This paper talks about the World of Web Scraper, Web scraping is related to web indexing, whose task is to index information on the web with the help of a bot or web crawler. Here the legal aspect, both positive and negative sides are taken into view. Some cases regarding the legal issues are also taken into account. The Web Scraper's designing principles and methods are contrasted, it tells how a working Scraper is designed. The implementation is divided into three parts: the Web Crawler to fetch the desired links, the data extractor to fetch the data from the links and storing that data into a csv file. The Python language is used for the implementation. On combining all these with the good knowledge of libraries and working experience, we can have a fully-fledged Scraper. Due to a vast community and library support for Python and the beauty of coding style of python language, it is most suitable for Scraping data from Websites.

***Keywords* – Web Scraping; Screen Scraping; Web Crawling; Implementing Web Scrape; Designing Web Scraper**

I INTRODUCTION

We all use Web Browser to extract the needed information from the Web Sites, if you think this is the only way to access information from internet then you are missing out a huge range of available possibilities. Web Scraper is one of those possibilities, in which we are accessing the information on the internet using some programs and pre written libraries.

[1]Wikipedia Says "Web scraping (web harvesting or web data extraction) is a computer software technique of extracting information from websites. Usually, such software programs simulate human exploration of the World Wide Web by either implementing low-level Hypertext Transfer Protocol (HTTP), or embedding a fully-fledged web browser, such as Mozilla Firefox. Web scraping is closely related to web indexing, which indexes information on the web using a bot or web crawler and is a universal technique adopted by most search engines. In contrast, web scraping focuses more on the transformation of unstructured data on the web, typically in HTML format, into structured data that can be stored and analyzed in a central local database or spreadsheet. Web scraping is also related to web automation, which simulates human browsing using computer software. Uses of web scraping include online price comparison, contact scraping, weather data monitoring, website change detection, research, web mashup and web data integration."

[2] Web Harvy says "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. Data displayed by most websites can only be viewed using a web browser. Examples are data listings at yellowpages' directories, real estate sites, social networks, industrial inventory, online shopping sites, contact databases etc. Most websites do not offer the functionality to save a copy of the data which they display to your computer. The only option then is to manually copy and paste the data displayed by the website in your browser to a local file in your computer - a very tedious job which can take many hours or sometimes days to complete."

The concept of Web Scraping is not new to us, it is getting more famous these days because of the new Startups, as they don't have to do much hard work to get the data, they mostly prefer to use the data scraped from other similar websites and then they modify it as per their need. Due to this, the bigger existing companies are facing much loss as their data being anonymously gathered and then reproduced by some other companies.

II IS WEB SCRAPING LEGAL?

This question is always left unanswered properly. There are lots of different views of different people on the legal and illegal aspects of Scraping the Web. In today's world we can see many examples of the legal use of Web Scraper such as price comparison websites and reviewing Websites. In this section we will see some example cases and try to answer this question.

[3] The Ebay's action "Not much could be done about the practice until in 2000 eBay filed a preliminary injunction against Bidder's Edge. In the injunction eBay claimed that the use of bots on the site, against the will of the company violated Trespass to Chattels law. The court granted the injunction because users had to opt in and agree to the terms of service on the site and that a large number of bots could be disruptive to eBay's computer systems. The lawsuit was settled out of court so it all never came to a head but the legal precedent was set." Another case was seen "In 2001 however, a travel agency sued a competitor who had "scraped" its prices from its Web site to help the rival set its own prices. The judge ruled that the fact that this scraping was not welcomed by the site's owner was not sufficient to make it "unauthorized access" for the purpose of federal hacking laws." Facebook's action "In 2009 Facebook won one of the first copyright suits against a web scraper. This laid the groundwork for numerous

lawsuits that tie any web scraping with a direct copyright violation and very clear monetary damages.” There is a case of AT&T’s where “Andrew Auernheimer was convicted of hacking based on the act of web scraping. Although the data was unprotected and publically available via AT&T’s website, the fact that he wrote web scrapers to harvest that data in mass amounted to “brute force attack”. He did not have to consent to terms of service to deploy his bots and conduct the web scraping. The data was not available for purchase. It wasn’t behind a login. He did not even financially gain from the aggregation of the data. Most importantly, it was buggy programing by AT&T that exposed this information in the first place. Yet Andrew was at fault. This isn’t just a civil suit anymore. This charge is a felony violation that is on par with hacking or denial of service attacks and carries up to a 15-year sentence for each charge.”

[4] A Quora user answered “The key part is what you want to do with the scraped data. If you use it for your own, personal use, then it is legal as it falls under fair use doctrine. There is nothing special in accessing data for yourself with a browser, you can use other means i.e. scraping.The complications start if you want to use scraped data for other, especially commercial, purposes. However even then you may be able to do a lot. A good example is deep linking, a practice where links to pages within a site are placed on another site, bypassing target site home page. There have been several legal precedents for such cases and in several of them courts have ruled that deep linking is legal, even including short descriptions and meta data from target pages, as long as it is clear that the site where deep links are placed is not claiming ownership of the data.”

The legality of the scrapper is still unclear based on the views of different people, the only thing we can still conclude about the legality is that if we are not doing any harm to the website and not selling the scrapped data then it is legal otherwise it is not any more illegal.

III DESIGNING A CUSTOM SCRAPPER

A Web Scrapper broadly composed of two parts they are:
Web crawler for crawling links + Data Extractor from crawled links.

A Web Crawler

[5] Wikipedia says “A Web crawler is an Internet bot which systematically browses the World Wide Web, typically for the purpose of Web indexing. A Web crawler may also be called a Web spider, an ant, an automatic indexer, or (in the FOAF software context) a Web scutter.”A Web Crawler usually crawls website using recursive algorithms, in which it scans the first page then finds the links on that page,stores them in a type of data structure and then fetches first links from the stored links, then open the webpage on that link and stores them into the same data structure and recursively repeats the process, till all links get crawled.

B Data Extractor

The Data Extractor extracts the needed information from the Web Page. There is lot of junk and useful data is mixed on the website, we have to look only for the needed useful information. To separate out the useful one from the mixture we target only that part of page in which information is present. To target that specific part, we are using the CSS Selectors.

IV WEB CRAWLER EXPLAINED

[5] Wikipedia explains “Web search engines and some other sites use Web crawling or spidering software to update their web content or indexes of others sites' web content. Web crawlers can copy all the pages they visit for later processing by a search engine which indexes the downloaded pages so the users can search much more efficiently.”

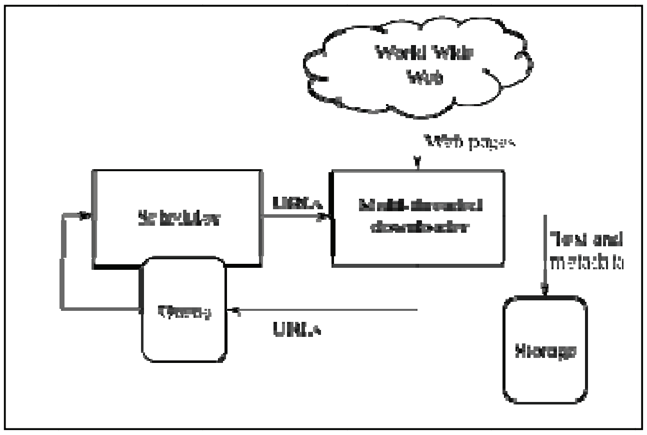


Fig. 1. Architechture of Web Scawler[5]

The working of web crawler is very simple, it starts with list of URLs to visit, that is called seeds. The crawler then visits all the URLs, then it processes all the web pages on those URLs, then the crawler finds the anchor tag on which links are contained, those links are stored in a list which will be further processed one by one and more links are fetched from them, that list of URL is called the crawl frontier. Two types of URLs are seen by crawler while visiting the web pages, one is Relative URL and another is Absolute URL. First the crawler has to identify weather an URL is Relative URL or Absolute URL, if it is Absolute URL then , the crawler can simply store those URL in the list and start fetching pages from those URL, and if the URL is Relative URL, then we need to provide the base URL, sometimes there is a case when base URL is different to get complete URL of the crawling links , there we have to do manual analysis of Web Page source, identifying the common pattern in the needed URL, then accordingly we convert them into Absolute URL.Sometimes there are redirect loops that can cause our crawler to get stuck in infinite loop if we have left the system and kept the crawler running, so these types of many problems are there while crawling so we need to take into account all these types of issues while designing a crawler. Let's take an example on Wiki related to one such issue "A simple online photo gallery may offer three options to users, as specified through HTTP GET parameters in the URL. If there exist four ways to sort images, three choices of thumbnail size, two file formats, and an option to disable user-provided content, then the same set of content can be accessed with 48 different URLs, all of which may be linked on the site. This mathematical combination creates a problem for crawlers, as they must sort through endless combinations of relatively minor scripted changes in order to retrieve unique content."

C Crawling Policy

Following are the combination of policies who gives thebehavior of a Web crawler:

Selection Policy - It States The Pages To Download.

Selection policy simply refers to prioritizing Web Pages based on metric of their importance. The importance in the Web Crawler to be designed for a Web Scrapper can be easily identified, as here we only have to focus some important links only not on all the links on that websites, as following all the links will result in wastage of time and resources while running the Web Crawler to find the links.

Re-Visits Policy -It States When To Check For Changes To The Pages.

In today's world most websites are found to have dynamic nature, suppose we have planned to crawl a website for all available links to start extracting data, it could take days or weeks to fully crawl a big website. The time at which our crawler finished getting the links the website had added new updated content, now we have to design the crawler in such a way that it can identify the new content itself and doesn't waste resources on the old content that has been already fetched. The check for changes part of crawler can add very much efficiency in writing a long term running crawler.

Politeness Policy - A Politeness Policy That States How To Avoid Overloading Web Sites.

This policy is used to prevent the crawler from overloading a website from sending multiple request so that the website doesn't goes offline, we need to consider that crawler can go anywhere on the website and a much higher speed than human do, this could affect the performance of the website. The solution could be using robots.txt exclusion in which the administrator specifies which URL are not to be crawled by the crawler. To make our crawler faster we make a crawler that make multiple request at one time, but this will degrade the performance of the website for genuine user. As developer we can provide an artificial delay between requests so that there are not such issues to the websites while crawling them.

Parellelization Policy - It States How To Coordinate Distributed Web Crawlers.

In this policy we speed up the crawler by running parallel crawler side by side, with the goal to maximize the visiting rate of the website and avoiding repeated links from getting into database of links. Many algorithms can be designed to make the crawler to work parallel. And to avoid getting duplicate links we need to write the links into database as soon as they are found. Then for added layer filter we can check the new link with all already found links, but this will add some more complexity into the code.

D Crawling The Deep Web

Deep Web is the part of internet that is not indexed by even massive search engines such as google. The surface web is just a small part of whole internet we have. The sites on Deep Web contain much more information than Surface Web. So sometimes we need to crawl and extract the data from Deep Web too. But the structure of websites on Deep Web is much challenging, so we need to design our crawler by doing careful analysis of the structure of website we are focusing on. Websites placed on Deep Web often goes offline, some websites are there which comes online at specific time, so to extract content from such website we need to take into account lots of issues that can be faced by us.

Browser Identification

Using the USER AGENT field, the HTTP Headers the Web Crawler is identified to a Web Server. Web Crawler identification is helpful for Web Administrators to communicate to the Crawler owner. But some website doesn't allow those crawlers which don't identify themselves as some famous Web Browsers, so for these the Crawlers USER AGENT field is spoofed to a Web Browser so that crawler can do its work easily. This spoofing sometimes prevents Crawl traps. So we have to pose like a genuine user.

V DATA EXTRACTOR

The Crawled List of links now available to us in which we are going to find the required information. If we start to gather whole website and save it as a copy, this won't be that much useful. So we have to extract the useful information and convert that into needed format. All these are done step by step.

i. We find out the content we are going to approach; this can be done using the inspect element button provided in the context menu of majority of Web Browser.

ii. We will find out the common pattern of CSS Selectors which is same on all other similar pages.

iii. Check out which method you are going to use you have support for Xpath or Simple CSS Selectors, depending upon we can carry our work further.

iv. We have extractor ready, but we have to store the data in a well formatted manner which should be very common to get converted into any other format very easily, one such format is CSV, check for the support of CSV into your programming language library.

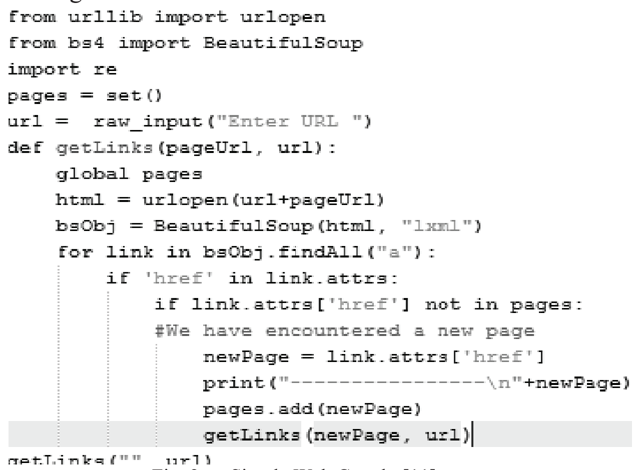
v. We have Data Extractor ready, do a recheck by extracting content on the temporary file so that CSV format gets a well formatted required output.

Now we have required data, this data can be easily converted into any other format as CSV is so common to all. We can convert it into JSON or SQL format.

VI IMPLEMENTAION OF CRAWLER USING PYTHON

[11] In this implementation we are using PYTHON as coding language, the reason behind choosing PYTHON is that PYTHON has vast community support and good amount of libraries available for crawling a website such as its inbuilt urllib, third party library such as mechanize, etc.

In the following code we have imported libraries named urlopen from urllib, urlopen will help us in visiting the URL by opening them,passing a URL to urlopen will do that. Then we have imported BeautifulSoup from bs4, BeautifulSoup is very powerful library for creating tree like structure of the content of web page. Then we have used python set, a set in Python have a special property that it can have list elements but all the elements stored will be unique regardless of their position, set are unordered list in Python. Then we have created a function named get_links to get the links from the website, in this function we pass the fetched URL and the URL as base URL, when the function will be called first time the fetched pageUrl variable will be given empty string as we don't have any URL fetched now, then after the recursion calls we have passed the fetched link in the pageUrl as now we have the URL. In the get_links function, we are converting the relative URL to absolute URL by concatenating the fetched URL and the base given provided URL, we have fetched all the links then using findAll we find the "a" tag in the page then we check for "href" in the attributes list of "a" tag. Then we check if that link is in that set or not, if link is not in the set then store it there, if it is in the set then ignore it and using recursive algorithm we pass the next link in the set to this function to keep fetching the links and storing them in the set.

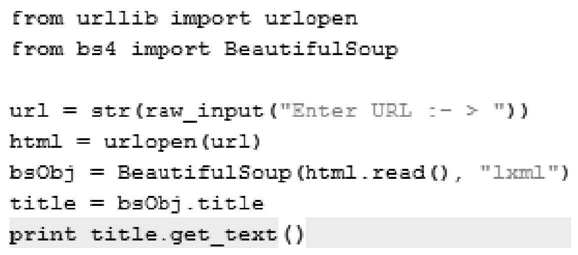


```
from urllib import urlopen
from bs4 import BeautifulSoup
import re
pages = set()
url =  raw_input("Enter URL ")
def getLinks(pageUrl, url):
    global pages
    html = urlopen(url+pageUrl)
    bsObj = BeautifulSoup(html, "lxml")
    for link in bsObj.findAll("a"):
        if 'href' in link.attrs:
            if link.attrs['href'] not in pages:
            #We have encountered a new page
                newPage = link.attrs['href']
                print("----------------\n"+newPage)
                pages.add(newPage)
                getLinks(newPage, url)
getLinks("", url)
```

Fig. 2. . Simple Web Crawler[11]

VII IMPLEMENTAION OF EXTRACTOR USING PYTHON

Data can be extracted using doing simple HTML parsing by using python module named BeautifulSoup which is good in creating tree structure of html elements. Using tree structure of BeautifulSoup, we can navigate through the elements and reach the needed content easily.



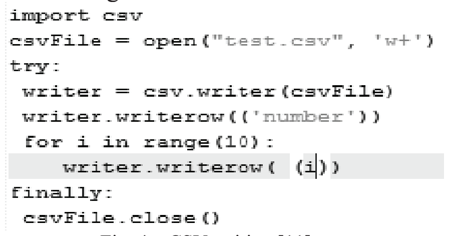
```
from urllib import urlopen
from bs4 import BeautifulSoup

url = str(raw_input("Enter URL :- > "))
html = urlopen(url)
bsObj = BeautifulSoup(html.read(), "lxml")
title = bsObj.title
print title.get_text()
```

Fig. 3. Title Extractor [11]

Here also we have used urlopen and BeautifulSoup library, we have discussed about them in Implementation of Crawler section. Now we are asking for the URL from the user using raw_input then opening the URL using urlopen we get the raw content, then that content is converted into tree structure using BeautifulSoup , then using BeautifulSoup we have fetched the title of the Web Site at that URL, in similar fashion we can print the needed information using CSS Selectors also , for more you have to read the documentation of BeautifulSoup, then on the title object we have used get_text function as title object has the html code, but we need only the text.

VIII WRITING TO CSV

PYTHON has great support for CSV, you just have to import the CSV module and you can use that with your inbuilt file handling system. CSV is a bit faster, smaller in size, very easy to handle (even in Excel) and many existing applications understand it, it is a widely used standard.Here we have used csv library available into Python, then we open a file name test.csv in append mode that will create and add the content into file, then we are writing into row the heading, then using for loop we are writing the number into the row.



```
import csv
csvFile = open("test.csv", 'w+')
try:
 writer = csv.writer(csvFile)
 writer.writerow(('number'))
 for i in range(10):
    writer.writerow( (i))
finally:
 csvFile.close()
```

Fig. 4. CSV writing [11]

IX FUTURE SCOPE

While throwing lights on the Future of Web Scraper it can be seen that people will start to develop many kind of services such as Price Comparison Websites, Big-Data Analysis. With Internet and web technology spreading, massive amounts of data will be accessible on the web. Since 'big data' can be both, structured and unstructured; web scrapers will have to get sharper and incisive. With the rise of open source languages like Python, R & Ruby, Customized scraping tools will only flourish bringing in a new wave of data collection and aggregation methods. The web scraping can be seen as providing solution to those who want data for their big data analysis.

X CONCLUSION

"If programming is magic, then web scraping is wizardry." says Ryan Mitchell, author of Book Web Scraping using Python. From the discussed topics we can conclude that the use of Scraper in the coming world will be increased significantly. As Scraper opens up another world of retrieving information without the use of API, and mostly it is anonymously accessed.

But the people who are doing Scraping should take into account that they are not breaking any kind of law which could make them liable for any offence. It should also be considered that the resources should not be consumed too much so that the target website is not able to produce content for the legit users.

REFERENCES

[1]. https://en.wikipedia.org/wiki/Web_scraping

[2]. https://www.webharvy.com/articles/what-is-web-scraping.html

[3]. http://resources.distilnetworks.com/h/i/53822104-is-web-scraping-illegal-depends-on-what-the-meaning-of-the-word-is-is/181642

[4]. https://www.quora.com/What-is-the-legality-of-web-scraping

[5]. https://en.wikipedia.org/wiki/Web_crawler

[6]. Kolari , P. and Joshi, A. , "Web mining : research and practice , Computing in Science & Engineering", IEEE Transactions on Knowledge and Data Engineering, vol. 6, no. 2,Vol. 6 , No. 4 , 2004

[7]. Malik , S.K. and Ravi , S.M. , "Information Extraction Using Web Usage Mining, Web Scrapping and Semantic Annotation,IEEE International Conference on Computational Intelligence and Communication Networks (CICN), 2011

[8]. Fister, I. ; Fong, S. ; Yan Zhuang, "Data Reconstruction of Abandoned Websites", IEEE 2nd International Symposiumon Computational and Business Intelligence (ISCBI), 2014

[9]. Quang Thai Le and Pishva, D. , "Application of Web Scraping and Google API service to optimize convenience stores distribution", 17th IEEE International Conference on Advanced Communication Technology (ICACT), 2015

[10]. Jaunt. Java – web Scraping and Automation,http://jaunt-api.com/jaunt-tutorial.htm

[11]. Ryan Mitchell – Web Scrapping Using Python, First Edition, Orilley, June 2015