**WEB CRAWLER**

Crawling the web has been an area of research and intersrt for long just because of the intrinsic value of the target data .The document retrieved in web search have been found to truthful in majority of cases .This has led to study for better understanding of the structure of web and the layout of documents so that more efficient algorithms can be developer for crawling

Utility of web crawlers

deep web

Web information beyond login pages .So

Social media

Concept of social media has come from the need of humans to connect with people of common interests. e.g keeping up with friends from school,college or workplaces

Social networking has been the studty of social sciences,computer sciences,statistics and economics long before the advent of internet social media platforms

Applications:

Epidemology,Espionage

Architecture of crawler

It is the process of capturing and extracting data from social networks, such as Facebook, Twitter and LinkedIn. This data let you sense consumer behavior, trends and sentiments.

To understand how social media [data scraping](http://www.multitechit.com.au/web-scraping-services.php) takes place, you should know that it runs on a piece of code. It is called scraper. As it runs, the “Get” query rolls out to extract the HTML data coming from Facebook or any other social channels.

Thereafter, algorithms analyse a string of symbols, either in natural language or computer language or models in the Document Object Model (DOM) structure. This parsing process determines nodes (an object representing a part of the document).  Then, it creates a node processor to show output in a normalized format. In simple words, the scraper comes into play, filtering through the data to pick up the requisite data sets. Once the requirement is fulfilled, the data is translated into a specific format.

In the nutshell, a code is tailored to:

**a.** Recognise unique HTML site structures

**b.**Extract  and transform data

**c.**Store the captured data

**d.**Extract data from APIs

Challenges

1. Getting Blocked

Linkedin websites have defined **User-agent**: \* or **Disallow:/** in robots.txt So any automated loginand scraping I generally forbidden

# Notice: The use of robots or other automated means to access LinkedIn without

# the express permission of LinkedIn is strictly prohibited.

# See https://www.linkedin.com/legal/user-agreement.

# LinkedIn may, in its discretion, permit certain automated access to certain LinkedIn pages,

# for the limited purpose of including content in approved publicly available search engines.

# If you would like to apply for permission to crawl LinkedIn, please email whitelist-crawl@linkedin.com.

# Any and all permitted crawling of LinkedIn is subject to LinkedIn's Crawling Terms and Conditions.

# See http://www.linkedin.com/legal/crawling-terms.

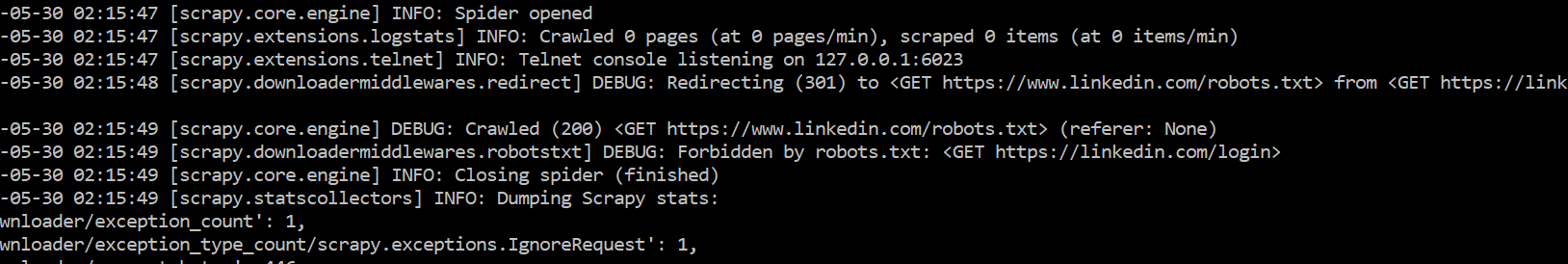
User-agent: Googlebot

Disallow: /addContacts\*

Disallow: /addressBookExport\*

Allow: /business/sales/blog\*

So when detected linkedin website denies scraping when any automated bot is detected in general



As we see here the login is getting redirected to robots.txt and get forbidden http status (403)

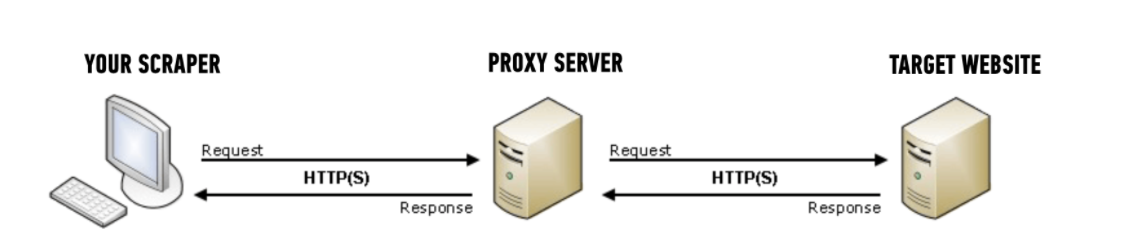
1. Scraping refused

Most popular websites have software to detect huge number of requests from one IP address

and blocking the ip suspecting DDOS attack

Solutions to prevent blocking:

Using Proxy server



Benefits of proxy server

Here are two main benefits to using proxies for your web scraping project:

1. Hiding your source machine’s IP address
2. Getting past rate limits on the target site

Many large sites have software in place to detect when there are a suspicious number of requests coming in from one IP address, since this usually indicates some sort of automated access – it could be scraping, or something security related like fuzzing.

The way this rate limiting software is usually setup, if too many requests come in from one IP address in a short amount of time, then the site will return some sort of error message to “block” future requests from that client for a pre-set period of time.

* faking user-agents

When your browser connects to a website, it includes a User-Agent field in its HTTP header. The contents of the user agent field vary from browser to browser. Each browser has its own, distinctive user agent. Essentially, a user agent is a way for a browser to say “Hi, I’m Mozilla Firefox on Windows” or “Hi, I’m google chrome on an iPhone” to a web server.

The web server can use this information to serve different web pages to different web browsers and different operating systems.

**Test scraper**

**Technologies used** :Python,BeautifulSoup(scraping library python),Selenium(testing automation )software/

Social media website :Linkedin

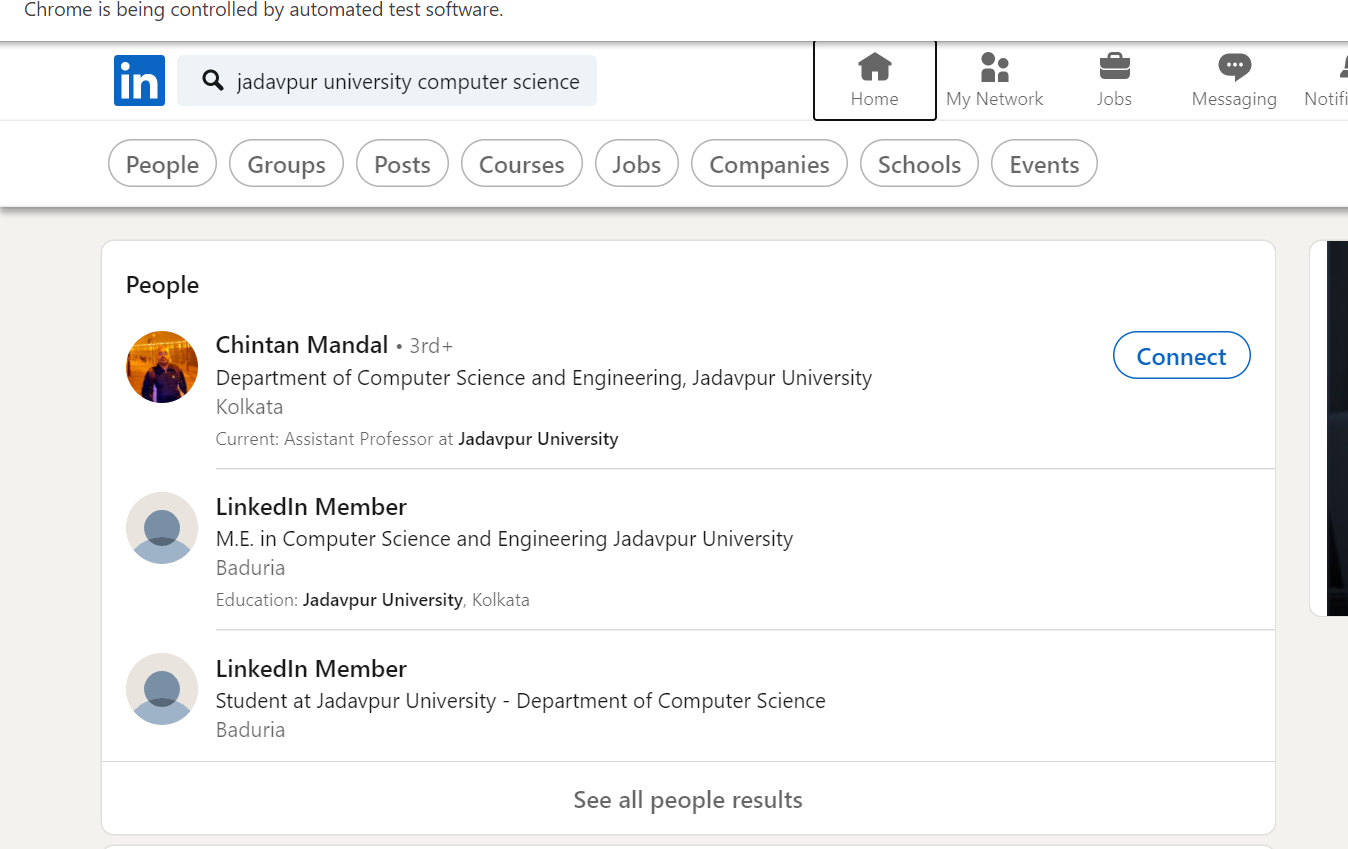
Objective of the crawler :To find alumni of any institute (focused crawler) “

Steps in the solution

1. Automatic logginng using selenium

2.Invoke a search by keying in the keywords for e.g “Jadavpur Unisversity Computer Science” and we

get the results like .T



3.

We get the DOM structure using developer tools and search for common patterns which represents

each profile in the search results like div class “entity-result\_\_content entity-result\_\_divider pt3 pb3 t-12 t-black—light”



4. Visit each href link for each person of interest

5. Like before extract all necessary attributes like name, year of passing of the specified institute and specified branch .The scraped data is put in a python dictionary which is later dumpled in an excel file or stored in database

**Metrics**

The scraper ran for around 25 mins scraping 159 profile and saved data for 59 profiles

**Challenges faced**

* Non uniform profiles

Grossly most profiles are non uniform . This is peculiarity of social media where no two profiles are\ same The length of a profile may not be same .

Some profile may have put a resume or som eresearch paper so I does not fit into template used for searching.

The root cause is the whole page is no getting loaded all the time for longer profiles.So the html file that is render may not contain all the web element s or scraping the required info.

Some profiles may educational details but experience details are not there which retun none so for the program to be resilient that had to be handled

* Some dynamic rendering is there using Javascipt which cannot be scraped
* Many have kept their profiles private so thye are simply showing as “Linkedin Member”

These profiles although visible in search results are thus not worth scraping as no usable information is gathered from them

* Changing web element properties

Linkedin keep changing the name of elements or class of web elements from time to time

So one approach of extracting a particular div of class name “education-section”

may not work few days later

* Slowness of program

The program ran in a single thread looking up profiles one by one . So it consumed time more so because of the delays introduced for politeness and to prevent from getting blocked

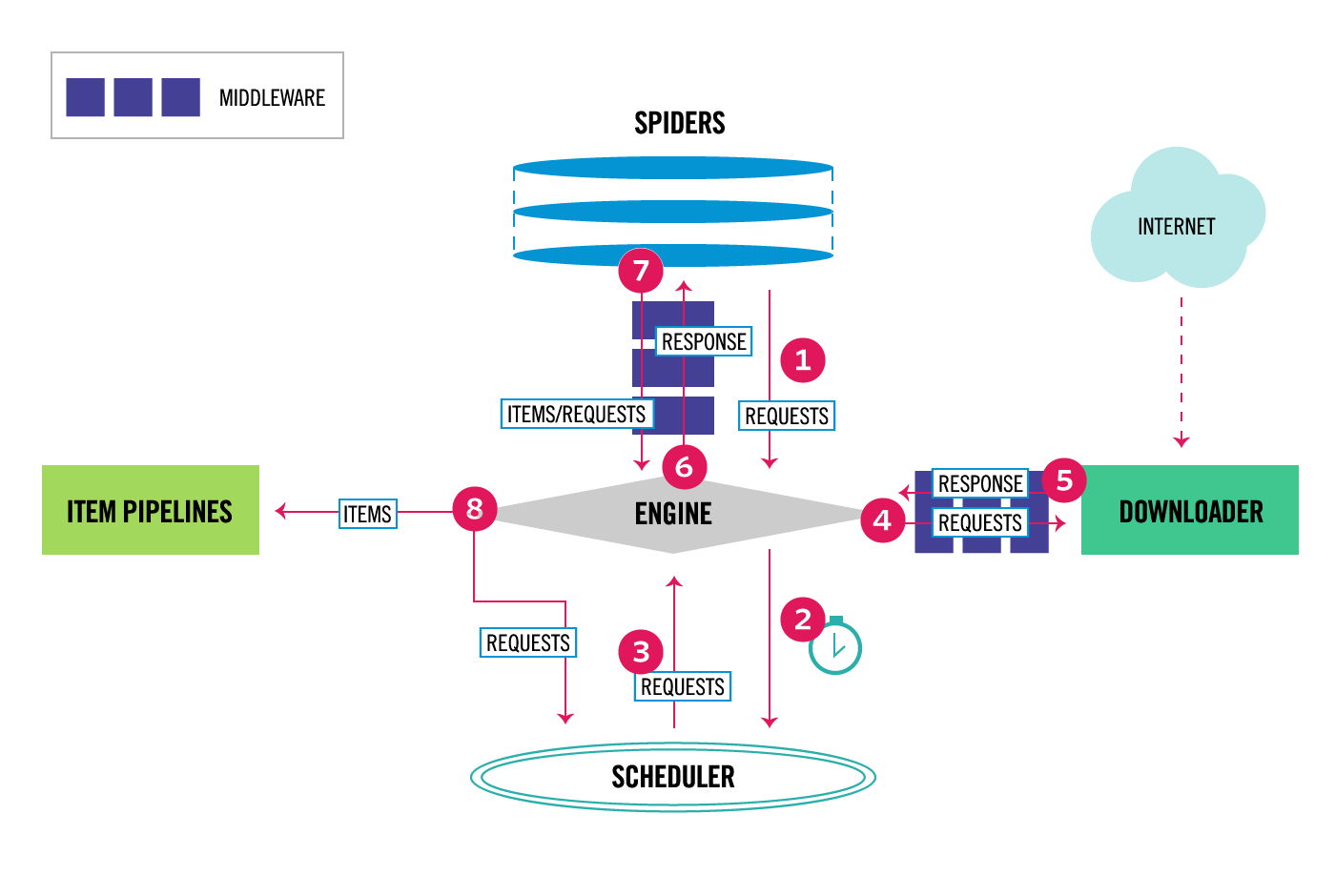
* Freshness of data

Profiles keep changing as people change their education or experience info.To achieve this

the profiles that are already scraped in one go needed to be scraped again.There are no other ways to determine change .Searche had to scheduled. But that posed a threat of being detected for visiting same links again

Other approaches tried

Scrapy :Using scrapy framework



References

* Social network theory (cf. Wasserman & Faust)