**WEB SCRAPPING TOOL**

**TECHNOLOGIES USED:**

* Python 3, Scrapy.
* Scrapy is a fast-high-level web crawling and web scraping framework, used to crawl websites and extract structured data from their pages.

**WHY SCRAPY and its BENEFITS:**

* The biggest feature is that it is built on Twisted, an asynchronous networking library, which makes the speed and performance of scraping better.
* Some websites like Facebook will be behind a login wall. Scrapy has built in form handling which you can setup to login to the websites before beginning your scrape.
* Scrapy is fault-tolerant, it lets you handle errors gracefully without interrupting the scraping if one requests fail
* Scrapy can do multiple requests at the same time being asynchronous which allows scraping runs to be much faster.
* Scrapy enables you to easily post-process any data you find. (it may have extra line breaks, extra commas in random places). Scrapy will let you handle these cases in a straight forward fashion.

**CHALLENGES:**

There were challenges in terms of technicality like:

1) Login into website form

2) Access to Facebook data,

3) dealing with JavaScript, ajax calls.

**SOLUTION:**

* Using Mobile-optimized website’ [m.facebook.com](http://mobile.facebook.com/) which doesn’t use any AJAX using Scrapy.

**SCRAPY ARCHITECTURE:**



**FLOW**:

* Credentials Input: Feed credentials (email id and password) to Scrapy Form Request, it will fill the form with credentials in start URL and returns home page.
* Crawl through page and new pages if any to fetch contents using Scrapy Request function.
* Fetch the scrapped data from response which request function sends to callback functions

**PERFORMANCE CHALLENGES:**

* Web scraping is a task that has to be performed responsibly so that it does not have a detrimental effect on the sites being scraped. Web Crawlers can retrieve data much quicker, in greater depth than humans, so bad scraping practices can have some impact on the performance of the site.
* Web scraping bots fetch data very fast, so, it is easy for a website to detect scraper as humans cannot browse that fast. If a website gets too many requests than it can handle, the website will easily detect you as a bot and it might become unresponsive.
* Hence, we have to make the scrapper look real, thus mimicking human actions by adding some download delays after crawling a small number of pages and choose the lowest number of concurrent requests possible. Even some random programmatic sleep calls in between requests can be used. Ideally put a delay of 10-20 seconds between clicks and not put much load on the website. Hence, due to this reason crawling and scrapping the page can experience a delay.
* To improve the scraping speed, Shared/Exclusive Proxy Services can be used. When scraping, your IP address can be tracked and traced. A web site will certainly know if one is crawling and collecting data. It could take data such as – user patterns or experience into account. Multiple requests from the same IP will lead scrapper to get blocked. So, when there are requests from a proxy machine, the target website will not know where the original IP is from, making the detection harder. Using a proxy allows crawling a website much more reliably. Significantly reducing the chances that your scrapper will get banned or blocked.
* Thus, when scraping the web at any reasonable scale with speed performance, using proxies is an absolute must.