Scrapping

PyData Salamanca

Andres Cevallos

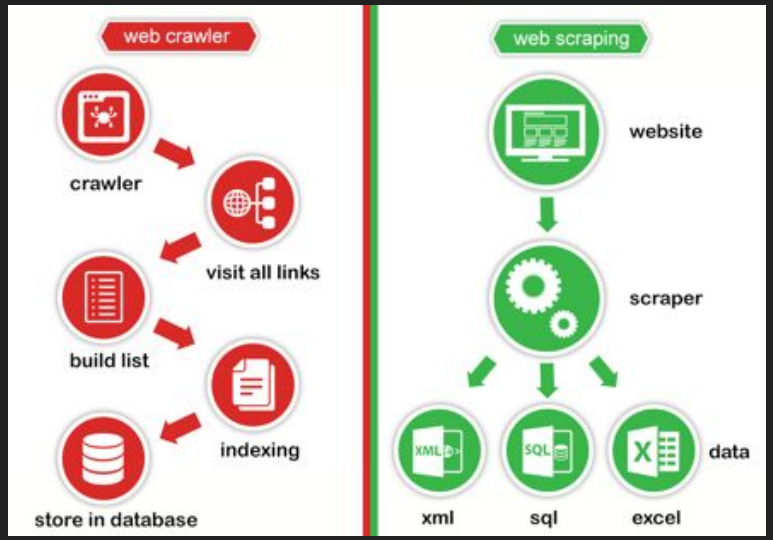
@variableisnull

https://github.com/andrescevp

Software Engineer at The Reptrak Company

What is what?

- Crawling: Extract information from the web
- Scrapping: Extract information from any source



Advantages, disadvantages and day to day

- Advantages
  - **Enrichable data**
  - No wait for API updates
  - “Unlimited” Access. All you can see can be extracted.
  - No rate limits
  - No freemium
  - “Anonimous” Access
  - *Reverse engineer is funny*
  - *With great power there must also come great responsibility!*

- Disadvantages
  - Sessions
  - Blocks (Ips, proxies)
  - Site changes (dynamic tokens, cookies)
  - Captchas
  - Easily finalize with a DDos
  - Waiting times
  - User tracking
  - ISP bans
  - Legal
  - Honeypots for scrappers

So… What I need?

- A good client - JS - dynamic.
- Some time researching and checking the website - cookies, interactions, etc
- Persistence
- HTTP (200 OK)
- DOM rules
  - AJAX, Scripts, Iframes, Broken code
- Captchas

Tooling

- Python
  - Request, Urllib, Time, BeautifulSoup
- Node
  - ExpressJS, Request, Cheerio
- PHP
  - Panther
- WebDrivers
  - Selenium
  - ChromeDriver/ FirefoxDriver
- Others
  - https://temp-mail.org/es/

- https://www.scraperapi.com/
  - Curl interface with Catpcha support
- https://www.scrapesimple.com/
  - Tell them what you want and get nack and CSV
- https://www.octoparse.com/
  - Scrapper UI (nice one)
- https://scrapy.org
  - Open pyhton framework with cloud.

Hands on!

- Linkedin: Get employee historical rotation
- Harris Poll: Getting data from Iframe + Compiled JS

Thanks!