Scrapy

* Section6
  + 18: Scrapy Architecture
    - Items.py: define items to output
    - Pipeline.py:
    - Setting.py: pipeline setting, robot\_obey, download delay
    - Avoid getting Banned!

It is very important to be careful while scraping websites; otherwise, you might be banned. Here are some tips to keep in mind while web scraping:

1- In the file settings.py activate the option DOWNLOAD\_DELAY or you can do that manually in your code through sleeping a for a random number of seconds. If you use sleep and random, here is a code snippet:

1. from time import sleep
2. import random
4. # Your code here
5. ...
7. sleep(random.randrange(1,3))

2- In the file settings.py activate the option USER\_AGENT like the following, or any Chrome or Firefox user agent [here](http://www.useragentstring.com/pages/useragentstring.php). Defining a user agent let you look more like a browser used by a real person, not an automatic robot.

USER\_AGENT = "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1"

3- Find external proxies and rotate IP addresses while scraping. You can use the package scrapy-proxies for the purpose. https://github.com/aivarsk/scrapy-proxies

4- For professional work, consider using ScrapingHub.com to host your scrapers - it offers a free limited plan.

* Section9: Deploying Scrapy spider to scraping Hub

**Recommendations Before Web Scraping:**

• Before web scraping, it is highly recommended to search for an API for the website you want to get data from. Most large websites offer APIs to make data extraction a better experience for both parties. So try first to search Google for an API for the website; if you find one, you do not need to scrape it. APIs generate JSON objects which are very similar to Python dictionaries, and from which data can be extracted using the Python JSON library.

• Before web scraping, it is highly recommended you read the Terms and Conditions of the website. Some websites clearly mention prohibiting web scraping without permission, or mention some legal or copyright aspects related to the use of its data.

• Before web scraping, employ common sense! Some web scraping or other robot activities are obviously illegal if they cause any direct or indirect damage to the company owning data or its customers. It is a good idea to discuss the purpose of a web scraping project with your client before accepting it.

• Before web scraping, prepare your code to be "polite": do not unnecessarily disable robots.txt of the website; space out your requests a bit so that you do not hammer the site's server; and it is better to run your spiders during off-peak traffic hours of the website.

* ScrapingHub
  + Free tool for scheduling spider
* Logging into Websites using scrappy
  + Use token
* Bulding Web crawler with scrappy
  + Use Rule
  + LinkExtractor
* Why/When We should use selenium
  + sraping thousand pages with Scrapy is 20 times faster than using Selenium. Furthermore, Scrapy consumes a lot less memory, and CPU usage than Selenium.

Section15

* + Scrapy Arguments
    - Define argument for \_\_init\_\_ function
    - Command for using arguments
      * Ex: scrapy crawl books -a category=”\*\*\*”
  + Scrapy Close function
    - Once the Spider is completed scraping so you can use it to analyze data, send data file via e-mail
    - Tips
      * Using os and glob module
  + Items
    - Use them when there is a lot of post-processing or cleaning work for data

Section16: Export Output to Files

* Exports to csv, json, xml using -o
* Exports to excel
  + Pip install openpyxl
* Pipeline
  + Pip install Pillow
* We can rename images downloaded