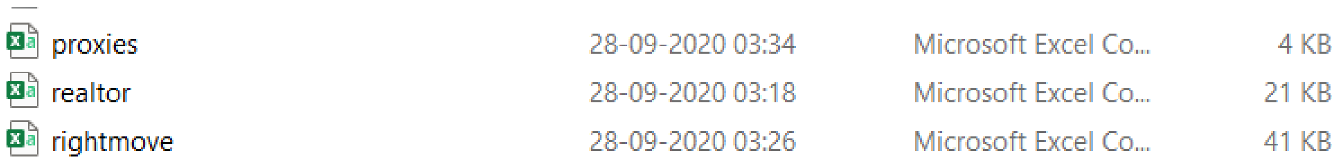
Project Name: Web-Scraping With Scrapy in 1 line

Table of Contents

Demo
Overview
Motivation
Technical Aspect
Installation
Run/How to Use/Steps
Directory Tree/Structure of Project
To Do/Future Scope
Technologies Used/System Requirements/Tech Stack
Credits

Demo



proxies	28-09-2020 03:34	Microsoft Excel Co...	4 KB
realtor	28-09-2020 03:18	Microsoft Excel Co...	21 KB
rightmove	28-09-2020 03:26	Microsoft Excel Co...	41 KB

Overview

This is a diving into Web Scraping with few lines of code.

Web Scraping 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.

It automates the gathering and dissemination of information. In the wrong hands, it can lead to theft of intellectual property or an unfair competitive edge. Therefore, before you scrape you need be careful and scrape only legal sites. Web scraping extracts underlying HTML code and, with it, data stored in a database. The scraper can then replicate entire website content elsewhere. Whether a website can be scraped or not, can check or know if a website allows scraping either by python or any tool or language, all you need do is to check the websites robots. Txt file by going to websiteName. Tld/robots.

This repository contains the code for Web Scraping using python’s various libraries.

It used scrapy and ipython libraries.

These libraries help to perform individually one particular functionality.

Scrapy is a Python framework for large scale web scraping. It gives you all the tools you need to efficiently extract data from websites, process them as you want, and store them in your preferred structure and format.

Ipython is a powerful interactive Python shell.

The purpose of creating this repository is to gain insights into how to scrape websites and collect data.

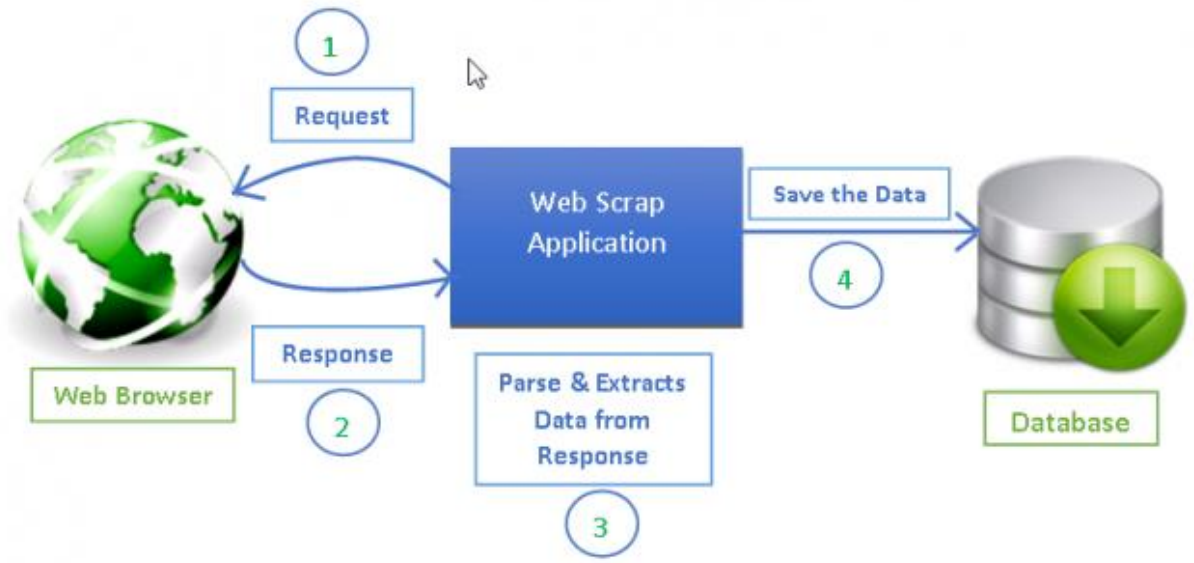
These python libraries raised knowledge in discovering these libraries with practical use of it. It leads to growth in my ML repository.
This above screenshot will help you to understand flow of output.

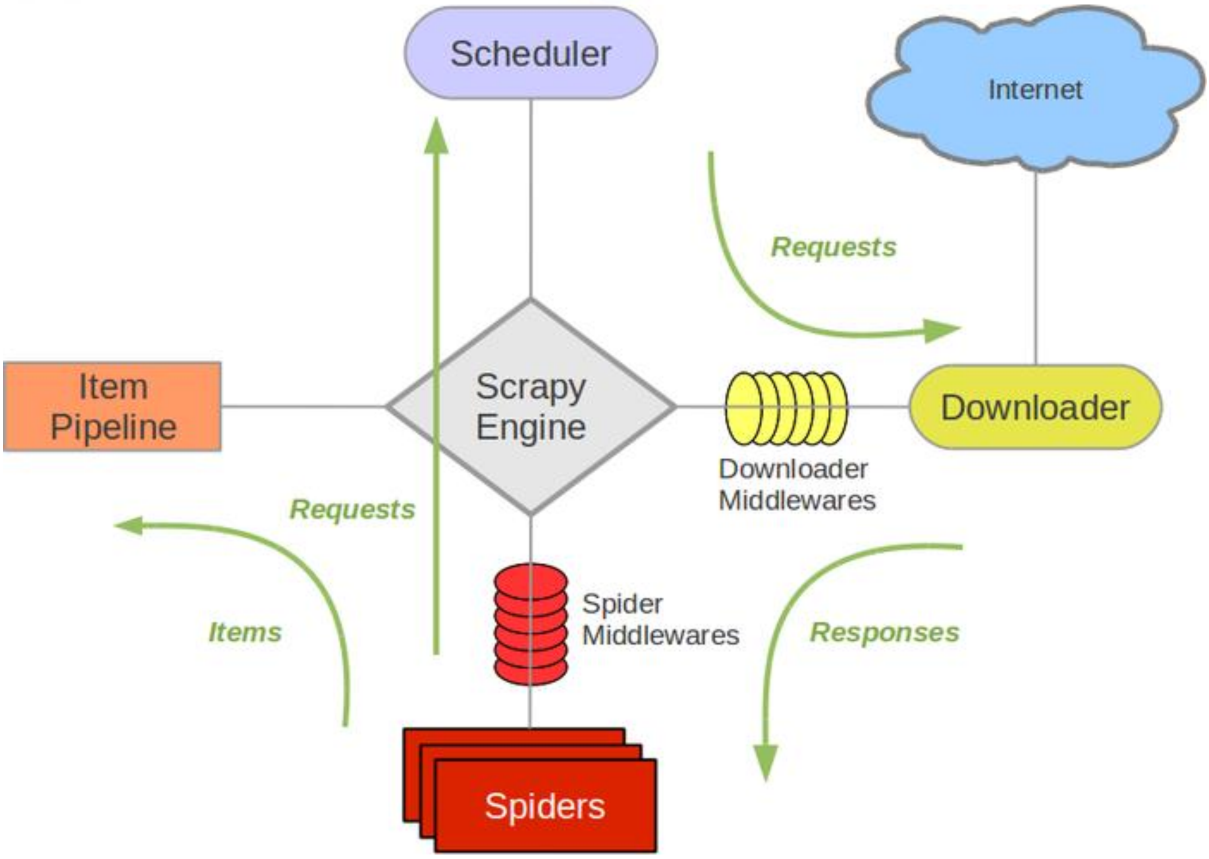
Motivation

Web-scraping provides one of the great tools to automate most of the things a human does while browsing. Web-scraping is used in an enterprise in a variety of ways – Data for Research, Products prices & popularity comparison, SEO Monitoring, Sales and Marketing. When we passed a html document or string to a beautifulsoup constructor, beautifulsoup basically converts a complex html page into different python objects. Basically, Web scraping is a process of automating the extraction of data in an efficient and fast way.
The reason behind building this is, for product-based company it is very important to continuously check on rating and feedback and improve it and also while building the new product it is important to carry out proper market research for the product. Since I am targeting product-based companies so it becomes one of the foremost aspects for me as well. Since I as IT Professional learnt to make websites during graduation years and now, I am learning to reverse engineer it by doing web scraping. So that completes the cycle. Web scraping is used in almost all industries be it journalism, finance, Data Science or E-Commerce. Web Scraping is core of market research and business strategies. Whether you want to start a new project or churn out a new strategy for an existing business, you need to invariably access and analyse a vast amount of data. This is where web scraping comes in. This concept caught my attention because for any business CRM is key aspect and web scraping for CRM becomes essential and that is where it opens up millions of opportunities. I have come across situation that; many times, it happens that company want to know about only particular information and not whole and here scraping helps to get exactly and only what you need and hence this way of collection of data is also part of many business decisions. One of the important goals of Web Scraping which encouraged me is, extraction of data by this method eliminates human error and therefore less outliers in data and final model prepared which led to 4/5th time of accuracy. Hence, I continue to gain knowledge while practicing the same and spread literary wings in tech-heaven.

Technical Aspect

Reason for working with scrapy in this project is because Scrapy is stronger and vast compared to that of Beautiful Soup. Scrapy will make the process of working around incomplete data much easier for you.





Installation

Using intel core i5 9th generation with NVIDIA GFORECE GTX1650.
Windows 10 Environment Used.
Already Installed Anaconda Navigator for Python 3.x
The Code is written in Python 3.8.
If you don't have Python installed then please install Anaconda Navigator from its official site.
If you are using a lower version of Python you can upgrade using the pip package, ensuring you have the latest version of pip, *python* -m pip install --*upgrade pip and press Enter.*

Run/How to Use/Steps

A virtual environment allows us to create an isolated working copy of python for a specific project without affecting the outside setup.
Keep your internet connection on while running or accessing files and throughout too.
Follow this when you want to perform from scratch.

Open Anaconda Prompt, Perform the following steps:

Creating Virtual Environment named “scrape”. You can give any name of your choice.

```
cd <PATH>
conda create -n scrape python=3.6
y
conda activate scrape
pip install scrapy
pip install ipython
```

Type:

```
scrapy
scrapy shell "URL"
```

code paste

That is, code of 1 line is as given in .py file after writing code only press enter to get output.

Then .csv file will get created in same working folder.

Do not Run .py file directly in anaconda prompt after installing libraries because it will not create appropriate content in file.

Correct way: copy code and run it in scrapy shell in-order to create .csv file.

Just type exit in shell to come out of shell and go back to prompt.

You can also create requirement.txt file as, pip freeze > requirements.txt

run files.

Creating Virtual Environment is necessary so that you do not have to install packages every-time you run the code. Once all required packages are installed in virtual environment then you only need to access/open the virtual environment and run the final file.

Follow this when you want to just perform on local machine.
Download ZIP File.
Right-Click on ZIP file in download section and select Extract file option, which will unzip file.
Move unzip folder to desired folder/location be it D drive or desktop etc.
Open Anaconda Prompt, write cd <PATH> and press Enter.

eg: cd C:\Users\Monica\Desktop\Projects\Python Projects 1\14)Web_Scraping\proj1 scraping using bs4+requests

Now, open virtual environment that you have created ie
conda activate scrape
In Anconda Prompt, pip install -r requirements.txt to install all packages.

Note: I have created scrape virtual environment and used for more than one project and therefore you might see more than one unused library in requirements.txt especially for this project so do not worry because I am using them in another project under similar virtual environment. Whenever you get No Module <name of package> Error then see its PyPI Documentation and Install it using pip install <package-name> written there. In some cases, you need to install its .whl file which I will inform you if its necessary.

Note: cd <PATH>
[Go to Folder where file is. Select the path from top and right-click and select copy option and paste it next to cd one space <path> and press enter, then you can access all files of that folder] [cd means change directory]

Directory Tree/Structure of Project

Folder: 14)Web_Scraping > proj4 scrapingin1linecode
Web_Scraping_4.py
Web_Scraping_5.py
Web_Scraping_6.py

To Do/Future Scope

Can try other website and products.

Technologies Used/System Requirements/Tech Stack

Scrapy

IP[y]: IPython

Credits

Code Monkey King Channel