User guide of HTML Scraper

Development & Running Environment

This html scraper Python script runs under Python 3. Beautiful Soup 4, urllib3 and lxml should be installed in the Python environment. You can install the libraries using 'pip' or you can download the source code to install. Please ensure that the libraries you installed is for Python 3.

Alternatively, if you do not want to change you real python environment, you can use PyCharm to help you develop this scraper. It is a great IDE and has the community version which is free to use. It provides virtual Python environment for each individual project and a package installer to help install the libraries. Please find File-> Settings-> Project: [project name]-> project Interpreter to create or configure your environment. You can install the libraries there easily with simple clicking the green plus.

How does the scraper work (desi market as an example)?

An example has provided in the current code. We aim to retrieval the information from https://www.desimarket.com.au about the companies' details. In order to reach this target, we developed this simple scraper. It has an input.csv file which stores the target URLs, an output.csv which stores the results and a main.py script (Figure 1).

Figure 1 Example structure

In that example, there are a list of companies (Figure 3) on the website and are divided into 20 pages. For the 20 pages, we store their URLs in the input.csv file (Figure 2) and will go through them line by line in the future.

	A	B	C	D	E	F	G
1	https://www.desimarket.com.au/find-indian-financial-services/6?pageNo=1						
2	https://www.desimarket.com.au/find-indian-financial-services/6?pageNo=2						
3	https://www.desimarket.com.au/find-indian-financial-services/6?pageNo=3						
4	https://www.desimarket.com.au/find-indian-financial-services/6?pageNo=4						
5	https://www.desimarket.com.au/find-indian-financial-services/6?pageNo=5						
6	https://www.desimarket.com.au/find-indian-financial-services/6?pageNo=6						

Figure 2 Input.csv

*In the example code, process_url method calls another method, process_sub_url, because the target information is in the subpages. In your own case, you may not need it.

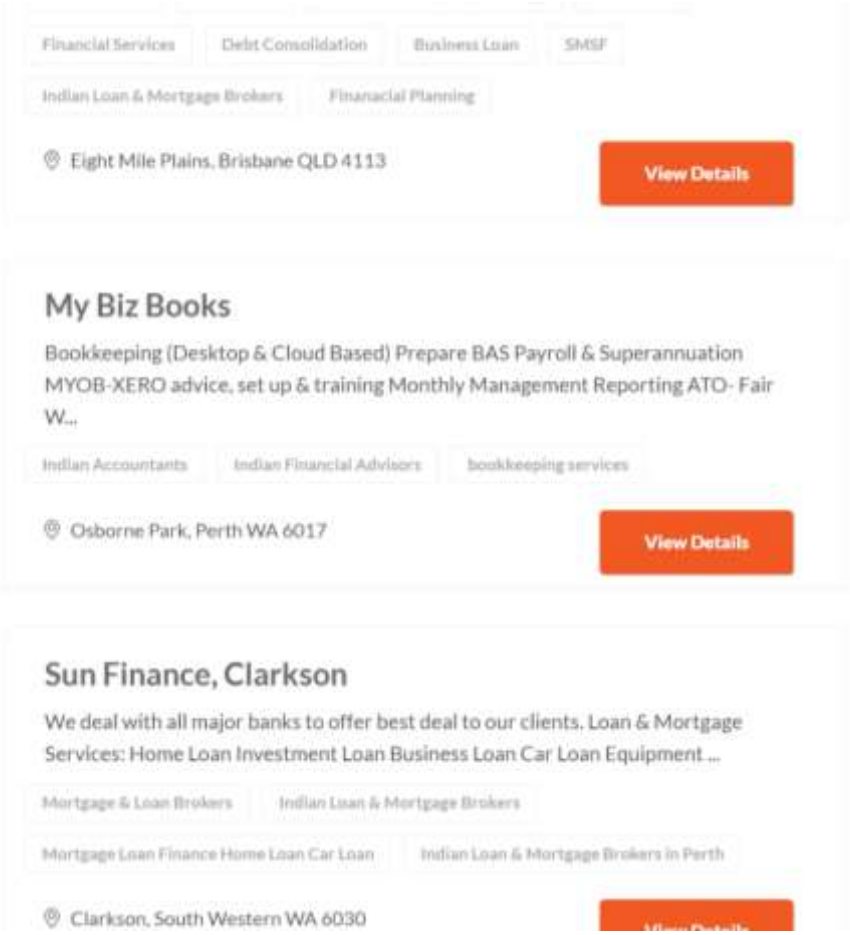

Figure 3 List of companies

For each page, we go through the list of the company links (Figure 3, *View Details* button). Each of them links to a page which contains the company details (Figure 4). So, we will get the information from them, including company name, address, phone number and the website.

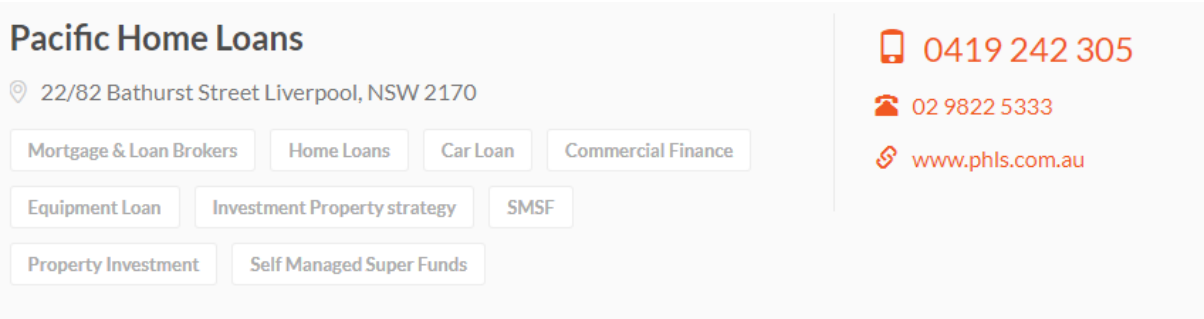

Figure 4 Company information page

The method used to locate the information we need is use the Beautiful Soup functions to locate the HTML tags, CSS classes or other contents. In our example, the company links are in the <h2> and <a> tag's "href" attribute (Figure 5), so we can access to all of them with the

*In the example code, process_url method calls another method, process_sub_url, because the target information is in the subpages. In your own case, you may not need it.

Beautiful Soup function find_all('h2'). Similarly, in the subpages of each company, we can use regular expressions to locate the phone numbers.

```
<div class="text box">

    <h2>
        <a href="/pacific-home-loans-liverpool-sydney-12965?SId=7585029">Pacific Home Loans</a>
    </h2>
    <div class="row">
        <div class="visible-xs">
            <div class="col-xs-8">
```

Figure 5 HTML source code

After we located the position of the target information, we need to store them into the CSV file in the required format. In this example, we write them into four columns, which are name, address, phone and website using the Python CSV dictionary writer.

	A	B	C	D	E	F
1	Pacific Home Loans, Liverpo	22/82 Bathurst Street Liverpool, NSW 2170	0419 242 305	www.phls.com.au		
2						
3	Devonshire Accountants	Shop 3 - 57 Emerlad Drive Regents Park, QLD 4118	0406 270 902	www.devonshireaccountants.co		
4						
5	C2C Accountants	Suite 2 Level 1 416 Logan Road Greenslopes, QLD 4120	07 3102 1357	www.c2caccountants.com.au		
6						
7	Tax Doctor Your Local Tax Ag	1/9 Flushcombe Road Blacktown, NSW 2148	0433 107 356	http://www.taxdoc.com.au		
8						
9	Rajesh Kumar, Liberty Advis	Surfers Paradise, QLD 4217	0435 861 184	www.liberty.com.au		
10						
11	Business Mantra Chartered .	5B/ 44 Hutton Street Osborne Park, WA 6017	0410 755 603	www.businessmantra.com.au		
12						
13	Grow Money	1 Clunies Ross Court Eight Mile Plains, QLD 4113	0439 077 133	www.growmoney.com.au		
14						
15	My Biz Books	5b 44 Hutton Street Osborne Park, WA 6017	0410 755 603	www.mybizbooks.com.au		
16						
17	Sun Finance, Clarkson	22 Edenbrooke Tce Clarkson, WA 6030	0411 032 405	www.sunfinancewa.com.au		
18						
19	LSM & Co Chartered Accoun	186A Prospect Road Prospect, SA 5082	0433 911 197	http://www.lsmco.com.au		

Figure 6 output.csv

How to make it suitable for your URLs?

To make the code suitable for your target webpages, you may follow the steps below:

1. List your target website in input.csv and make sure that each line only contains one URL.
2. Figure out the information you need and the format you want it to be.
3. View the HTML sources code of the target website and find out how to locate the information that you need. You can use the HTML tags, the CSS class names or the regular expressions. Please refer to Beautiful Soup Documentation for more information.
4. Modify the process_url( )* method to ensure that it return the correct information.
5. Modify the url_write( ) method to make sure the results are written into the file with the format that you want. The python CSV dictionary writer is recommended.
6. Run the script to get the result.

For more information, please find the comments in the source code.

*In the example code, process_url method calls another method, process_sub_url, because the target information is in the subpages. In your own case, you may not need it.

Skills required

- Python 3 programming
- HTML and CSS
- Regular Expression

Useful Links

- Python pip: https://docs.python.org/3/installing/
- PyCharm: https://www.jetbrains.com/pycharm/
- Urllib3: https://urllib3.readthedocs.io/en/latest/
- lxml: http://lxml.de/installation.html
- CSV read and write: https://docs.python.org/3/library/csv.html
- Regular expression: https://docs.python.org/3/library/re.html
- Beautiful Soup: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

*In the example code, process_url method calls another method, process_sub_url, because the target information is in the subpages. In your own case, you may not need it.