Sumeet Haldipur
2019130018
TE COMPS

EXPERIMENT 9

Aim:
To create web crawlers to scrape data from websites.

Requirements:
Python, Scrapy library

Problem Statement:
To scrape medal data from Olympic website.

Theory:
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.

1) **Spiders:**
   Spiders are classes which define how a certain site (or a group of sites) will be scraped, including how to perform the crawl (i.e. follow links) and how to extract structured data from their pages (i.e. scraping items). In other words, Spiders are the place where you define the custom behaviour for crawling and parsing pages for a particular site (or, in some cases, a group of sites).

2) **Selectors:**
   Scrapy comes with its own mechanism for extracting data. They're called selectors because they "select" certain parts of the HTML document specified either by XPath or CSS expressions. XPath is a language for selecting nodes in XML documents, which can also be used with HTML. CSS is a language for applying styles to HTML documents. It defines selectors to associate those styles with specific HTML elements.

3) **Items:**
   The main goal in scraping is to extract structured data from unstructured sources, typically, web pages. Spiders may return the extracted data as *items*, Python objects that define key-value pairs.

4) **Pipeline:**

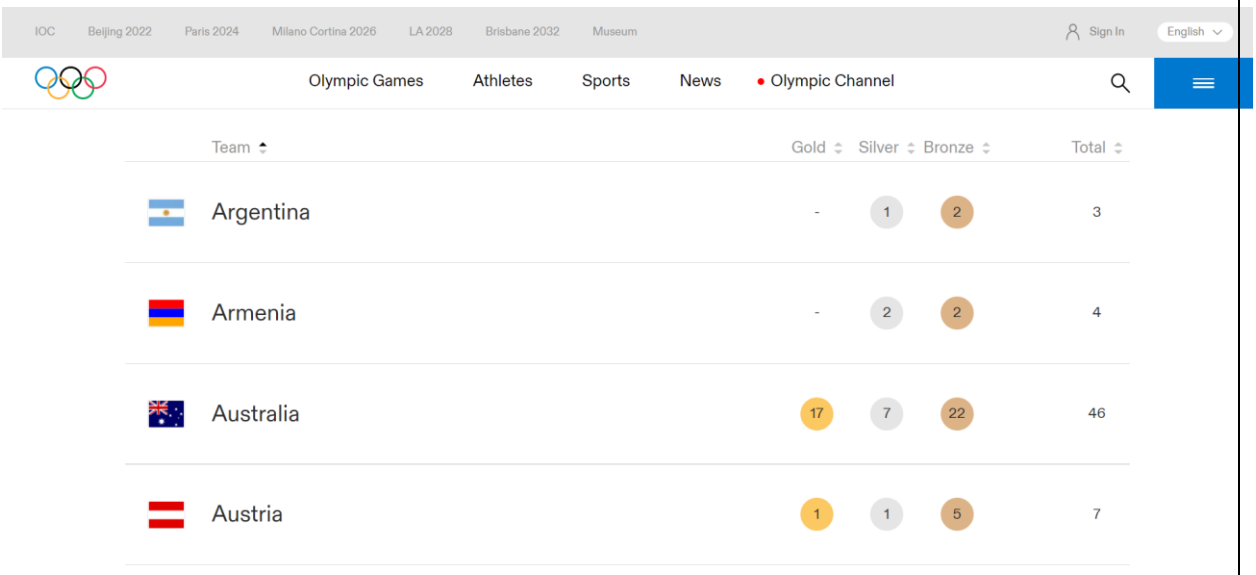
After an item has been scraped by a spider, it is sent to the Item Pipeline which processes it through several components that are executed sequentially. Each item pipeline component (sometimes referred as just “Item Pipeline”) is a Python class that implements a simple method. They receive an item and perform an action over it, also deciding if the item should continue through the pipeline or be dropped and no longer processed.

Typical uses of item pipelines are:

- cleansing HTML data
- validating scraped data (checking that the items contain certain fields)
- checking for duplicates (and dropping them)
- storing the scraped item in a database

Implementation:

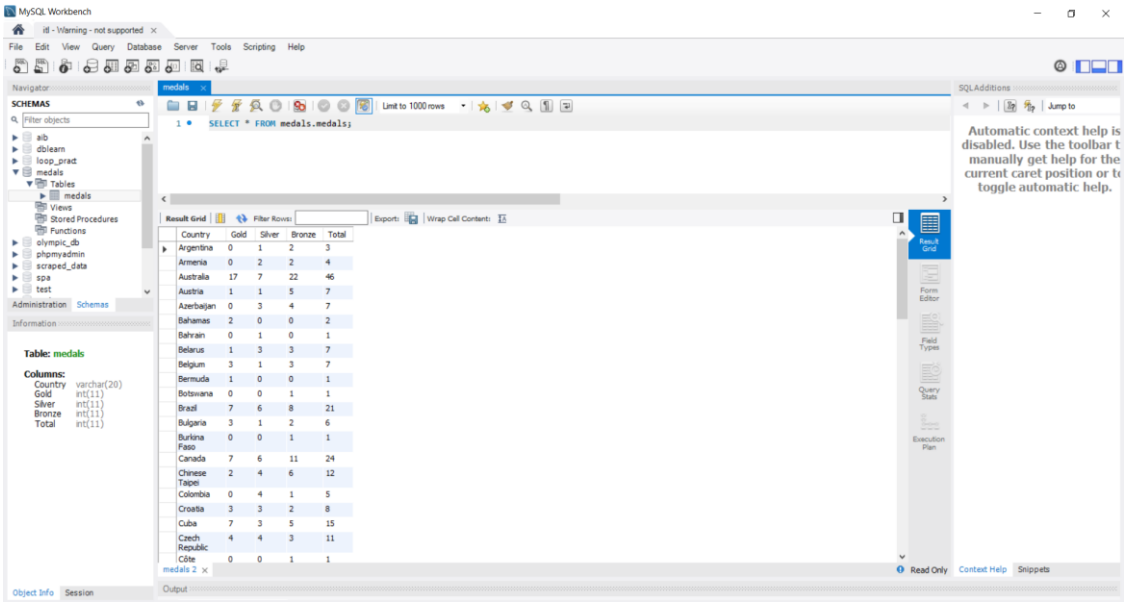
1. Web Page:



2. Scraping

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
2021-12-07 19:33:33 [scrapy.core.engine] INFO: Spider opened
2021-12-07 19:33:33 [scrapy.extensions.logstats] INFO: Crawled 0 pages (at 0 pages/min), scraped 0 items (at 0 items/min)
2021-12-07 19:33:33 [scrapy.extensions.telnet] INFO: Telnet console listening on 127.0.0.1:6023
2021-12-07 19:33:33 [scrapy.core.engine] DEBUG: Crawled (200) <GET https://olympics.com/robots.txt> (referer: None)
2021-12-07 19:33:34 [scrapy.core.engine] DEBUG: Crawled (200) <GET https://olympics.com/en/olympic-games/tokyo-2020/medals> (referer: None)
2021-12-07 19:33:35 [scrapy.core.scraper] DEBUG: Scraped from <200 https://olympics.com/en/olympic-games/tokyo-2020/medals>
{'Bronze': '2',
 'Country': 'Argentina',
 'Gold': '-',
 'Silver': '1',
 'Total': '3'}
2021-12-07 19:33:35 [scrapy.core.scraper] DEBUG: Scraped from <200 https://olympics.com/en/olympic-games/tokyo-2020/medals>
{'Bronze': '2', 'Country': 'Armenia', 'Gold': '-', 'Silver': '2', 'Total': '4'}
2021-12-07 19:33:35 [scrapy.core.scraper] DEBUG: Scraped from <200 https://olympics.com/en/olympic-games/tokyo-2020/medals>
{'Bronze': '22',
 'Country': 'Australia',
 'Gold': '17',
 'Silver': '7',
 'Total': '46'}
2021-12-07 19:33:35 [scrapy.core.scraper] DEBUG: Scraped from <200 https://olympics.com/en/olympic-games/tokyo-2020/medals>
{'Bronze': '5', 'Country': 'Austria', 'Gold': '1', 'Silver': '1', 'Total': '7'}
2021-12-07 19:33:35 [scrapy.core.scraper] DEBUG: Scraped from <200 https://olympics.com/en/olympic-games/tokyo-2020/medals>
{'Bronze': '4',
 'Country': 'Azerbaijan',
 'Gold': '-',
 'Silver': '3',
 'Total': '7'}
2021-12-07 19:33:35 [scrapy.core.scraper] DEBUG: Scraped from <200 https://olympics.com/en/olympic-games/tokyo-2020/medals>
{'Bronze': '-', 'Country': 'Bahamas', 'Gold': '2', 'Silver': '-', 'Total': '2'}
2021-12-07 19:33:35 [scrapy.core.scraper] DEBUG: Scraped from <200 https://olympics.com/en/olympic-games/tokyo-2020/medals>
{'Bronze': '-', 'Country': 'Bahrain', 'Gold': '-', 'Silver': '1', 'Total': '1'}
2021-12-07 19:33:35 [scrapy.core.scraper] DEBUG: Scraped from <200 https://olympics.com/en/olympic-games/tokyo-2020/medals>
{'Bronze': '3', 'Country': 'Belarus', 'Gold': '1', 'Silver': '3', 'Total': '7'}
2021-12-07 19:33:35 [scrapy.core.scraper] DEBUG: Scraped from <200 https://olympics.com/en/olympic-games/tokyo-2020/medals>
{'Bronze': '3', 'Country': 'Belgium', 'Gold': '3', 'Silver': '1', 'Total': '7'}
2021-12-07 19:33:35 [scrapy.core.scraper] DEBUG: Scraped from <200 https://olympics.com/en/olympic-games/tokyo-2020/medals>
{'Bronze': '-', 'Country': 'Bermuda', 'Gold': '1', 'Silver': '-', 'Total': '1'}
2021-12-07 19:33:35 [scrapy.core.scraper] DEBUG: Scraped from <200 https://olympics.com/en/olympic-games/tokyo-2020/medals>
```

3. Data stored in database



Country	Gold	Silver	Bronze	Total
Argentina	0	1	2	3
Armenia	0	2	2	4
Australia	17	7	22	46
Austria	1	1	5	7
Azerbaijan	0	3	4	7
Bahamas	2	0	0	2
Bahrain	0	1	0	1
Belarus	1	3	3	7
Belgium	3	1	3	7
Bermuda	1	0	0	1
Botswana	0	0	1	1
Brazil	7	6	8	21
Bulgaria	3	1	2	6
Burkina Faso	0	0	1	1
Canada	7	6	11	24
Chinese Taipei	2	4	6	12
Colombia	0	4	1	5
Croatia	3	3	2	8
Cuba	7	3	5	15
Czech Republic	4	4	3	11
Côte	0	0	1	1

Conclusion:

From the above experiment, I have learned the following:

- Knowledge about scrapy library of python.
- Hands on experience of creating web crawlers to scrape data from websites.
- Storing data extracted from crawlers into a database

References:

1. https://docs.scrapy.org/en/latest/
2. https://www.analyticsvidhya.com/blog/2017/07/web-scraping-in-python-using-scrapy/
3. https://pypi.org/project/Scrapy/
4. https://www.tutorialspoint.com/scrapy/index.htm
5. https://www.datacamp.com/community/tutorials/making-web-crawlers-scrapy-python