Information Retrieval

Crawling IEEE

Kourosh Hassanzadeh
Mohammad Hesam Ghasemi
Alireza Sajjadi

June 2024

Load IEEE

- First, we load the IEEE website and then search for the ***Blockchain*** expression.
    - To do this, we copy the Xpath of search input on the main page of IEEE and then send "Blockchain" to it.
    - Then we wait for all components on the page to load.
    - After the list of articles loads, we get the href attribute of each article to iterate over them and open each one in a new tab.

def wait_page_load():

- This function waits until the last element of the page, which is the ***Feedback*** button, loads.

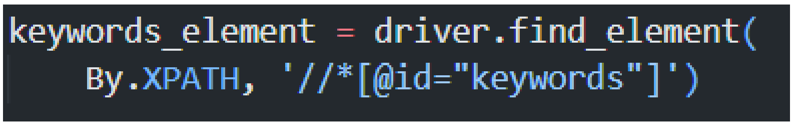
def get_links():

- This functions finds all elements with the class name ***fw-bold***. These are links of articles, so we save their href attribute to open each article in a new page in the *extract_article_info()* function.

def extract_article_info():

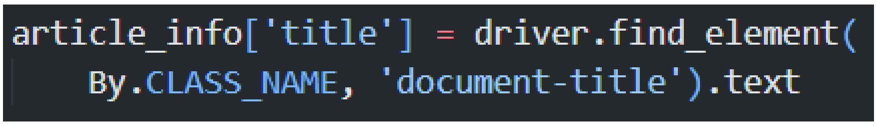
- This function gets all links of articles on a page, opens them in a new tab, and extracts all the necessary information.

- They are found by *Xpath* or *ClassName* or *CSS Selector*. Exapmle:



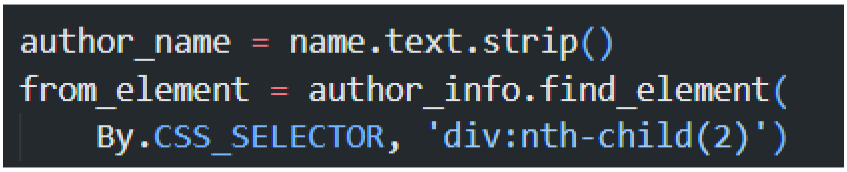
```
keywords_element = driver.find_element(
    By.XPATH, '//*[@id="keywords"]')
```

by Xpath



```
article_info['title'] = driver.find_element(
    By.CLASS_NAME, 'document-title').text
```

by ClassName



```
author_name = name.text.strip()
from_element = author_info.find_element(
    By.CSS_SELECTOR, 'div:nth-child(2)')
```

by CSS Selector

some problems we have faced:
  - Number of pages: At first, we didn't know that some articles doesn't have page numbers.
  - Citations: We had to search for a list of citations together. We found the element with the *document-banner-metric-count* class and then extracted *Cites in Papers* and *Cites in Patent* and *Full Text Views* from that element.
  - Authors:
    1. Some authors are related to multiple companies, which made it difficult.
    2. When we found the related element to authors, it returned a string, and processing it was more difficult than processing a list.
    3. At first, we opened the ***Authors*** tab from the navigation bar on the left side of the page. This didn't work for all articles, and we found that we should click on the ***Authors*** button at the end of the page.
  - KeyWords:
    We had the same problem as the third problem with Authors.

def change_page():

- We wrote this function to change pages(pagination) and iterate over articles on each page.
  - We wanted 5 pages, so we have a while loop until our counter reaches 6. We change the page after each iteration by clicking on the button with the class ***f'stats-Pagination_{i}'***.
  - We had a problem at this stage: when we wanted to go back from the last article we opened to extract information about it, the ***driver.back()*** command didn't work.
  - Our solution was to open the articles in a new tab. After iterating over each page's articles, we close that new tab, return the execution control to the main tab, and then go to the next page (in the extract_article_info() function).
  - After that, we call *get_url()*, which was explained.
  - Then we call *extract_article_info()*, which was explained.

We performed all of these steps once for sorting the articles by ***relevance***. After that, we changed the sort option to ***newest*** by clicking on the related button, changing the content of the button, and clicking on it again. This re-sorted the articles, and we repeated the steps to extract information by the new sorting. At the end we save them into a json file.

All members' participation in all sections was 10/10.