

# Scrapper Documentation

## Overview

This scrapper is built using BeautifulSoup library, AsyncChromiumLoader and Html2TextTransformer. The given code is intended to extract and transform the content from various URLs, convert the HTML content to text, and store it in a JSON file.

## Requirements

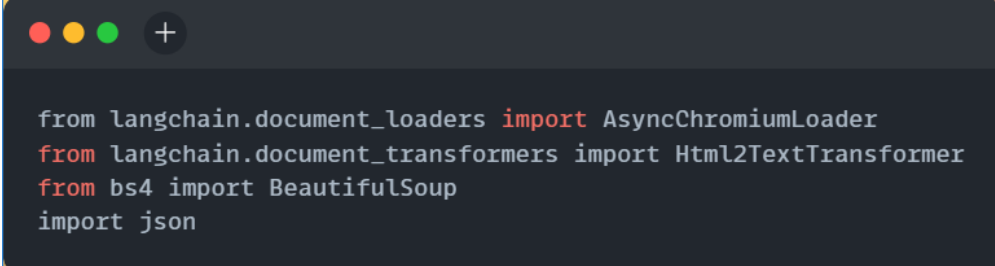
- All required modules are listed in the requirements.txt file. Install using the following command:-

```
pip install -r requirements.txt
```

## Usage

1. Navigate to the **scraper** directory and run the “scraper\_with\_bs.py” file to begin scrapping.

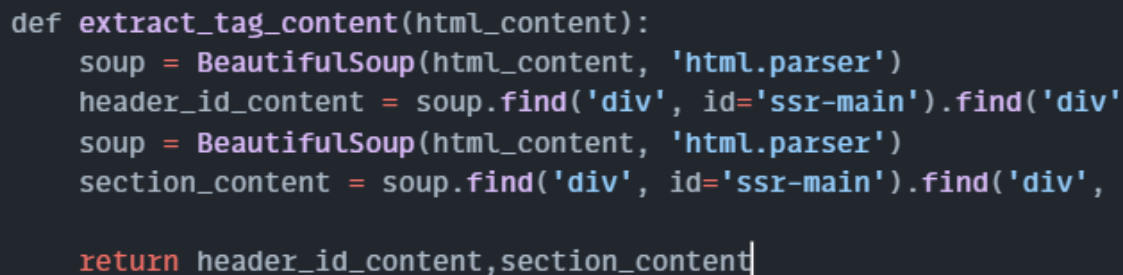
## Code Explanation



```
from langchain.document_loaders import AsyncChromiumLoader
from langchain.document_transformers import Html2TextTransformer
from bs4 import BeautifulSoup
import json
```

- Importing required modules.

## BeautifulSoup Implementation



```
def extract_tag_content(html_content):
    soup = BeautifulSoup(html_content, 'html.parser')
    header_id_content = soup.find('div', id='ssr-main').find('div')
    soup = BeautifulSoup(html_content, 'html.parser')
    section_content = soup.find('div', id='ssr-main').find('div',

    return header_id_content, section_content
```

- The function ‘extract\_tag\_content’. It takes HTML content, parses it with BeautifulSoup and extracts specific elements from the HTML structure. It returns two parts: ‘header\_id\_content’ from a <header> tag and ‘section\_content’ from a <div> within a specified section.

## Initializing AsyncChromiumLoader

```
all_data= []
output_file_path = "data.json"

for url in urlss:
    loader = AsyncChromiumLoader(urls=[url], headless=True, user_agent="USER_AGENT':'Desired Agent")

    html_documents = loader.load()
```

- This code initializes an empty list ‘all\_data’ and sets the path for a JSON output file. For each URL in the ‘urlss’ list, it creates an “AsyncChromiumLoader” object to load the webpage content with a specified user agent and headless browser settings. The loader.load() method retrieves the HTML documents from the given URL.

## Function Calling and Document Iteration

```
for html_doc in html_documents:
    html_content = html_doc.page_content

    data = extract_tag_content(html_content)
    new_page_content = ''.join(str(data))

    for doc in html_documents:
        doc.page_content = new_page_content
```

- This code iterates over the ‘html\_documents’ list, extracting and processing the content of each document using the “extract\_tag\_content” function. Finally, it updates each document's ‘page\_content’ in ‘html\_document’ with the combined ‘new\_page\_content’.

## HTML to Text conversion

```
html2text = Html2TextTransformer()
docs_transformed = html2text.transform_documents(html_documents)

content = docs_transformed[0].page_content[0: ]
md = docs_transformed[0].metadata
all_data.append({"page_content": content, "url": md})
print(url)
```

- It transforms HTML documents to text, extracts the content and metadata, appends them to a list, and prints the URL.

## **Notes**

- Modify the “urlss” list with desired list of urls obtained from the crawler.