WEB SCRAPING AND SOCIAL MEDIA SCRAPING

Retake Project Report

Submitted By:

Mehmet Tiryaki

Introduction to Movie Data Scraping Project

The Movie Data Scraping Project is a comprehensive web scraping solution aimed at extracting detailed information about movies from the "hdtoday.tv" website. This project combines the power of three popular Python libraries—Scrapy, BeautifulSoup, and Selenium—to automate the data collection process. By harnessing the capabilities of these tools, the project efficiently retrieves movie data, including movie names, IMDb ratings, release dates, genres, casts, durations, and countries. The gathered data can be further analyzed, stored, or integrated into various applications.

Project Overview

The primary objective of the Movie Data Scraping Project is to simplify the task of collecting valuable information about movies available on "hdtoday.tv". The integration of Scrapy, BeautifulSoup, and Selenium provides a holistic approach to web scraping, catering to different aspects of the data collection process.

Key Features

Scrapy Framework: Leveraging the Scrapy framework, the project offers a structured and organized approach to web scraping. Scrapy allows for the creation of specialized spiders that navigate through web pages and gather data using XPath expressions.

BeautifulSoup Parsing: The project employs the BeautifulSoup library to parse HTML content and extract specific elements. BeautifulSoup's flexibility makes it ideal for intricate data extraction tasks, such as retrieving movie links and details.

Selenium Automation: Utilizing the Selenium library, the project automates the web browsing experience. It interacts with web pages in a human-like manner, rendering JavaScript content and collecting information that might be challenging to access using traditional parsing methods.

Comprehensive Data Extraction: The project targets relevant HTML elements on movie pages to extract key details, including movie names, IMDb ratings, release dates, genres, casts, durations, and countries. The data is extracted using XPath expressions, regular expressions, and Selenium's find_element methods.

Structured Output: Extracted data is meticulously cleaned and structured into meaningful formats. This ensures that the collected information is easily manageable, shareable, and ready for further analysis.

User Customization: The project allows users to tailor their scraping experience by adjusting parameters such as page ranges, enhancing the flexibility and adaptability of the data collection process.

Usage

To utilize the Movie Data Scraping Project, follow the provided instructions for setting up the necessary dependencies and environment. The project includes spiders and scripts for Scrapy, Beautiful Soup, and Selenium, each serving distinct data scraping purposes. Running these components generates datasets containing comprehensive movie information from "hdtoday.tv".

Usage of BeautifulSoup Script

The BeautifulSoup script allows you to scrape movie details from the "hdtoday.tv" website. To use the script:

1. Ensure you have the required libraries (requests, pandas, re, and bs4) installed in your Python environment.
2. Copy and paste the script into your Python file.
3. Adjust the page_range variable to specify the range of pages you want to scrape (e.g., (1, 2)).
4. Run the script.
5. The script will scrape movie details based on the specified page range and create a pandas DataFrame containing the extracted data.
6. The resulting DataFrame will be printed, containing columns such as 'Movie_Name', 'Release_Date', 'Genres', 'Casts', 'Duration', 'Country', and 'IMDB_Rating'.

Usage of Selenium Script

The Selenium script automates the process of scraping movie data using a web browser. To use the script:

1. Ensure you have the Chrome web driver installed and the webdriver, pandas, re, and bs4 libraries installed in your Python environment.
2. Copy and paste the script into your Python file.
3. Adjust the page_range variable to specify the range of pages you want to scrape (e.g., (1, 2)).
4. Replace the paths to the Chrome web driver and the uBlock Origin extension in the script with the paths on your machine.
5. Run the script.

6. The script will open a Chrome browser window, navigate through the pages, scrape movie details, and create a pandas DataFrame containing the extracted data.
7. The resulting DataFrame will be printed, containing columns similar to those in the BeautifulSoup script.

Usage of Scrapy Spider

The Scrapy spider is designed to collect movie links and data from the "hdtoday.tv" website. To use the Scrapy spider:

1. Ensure you have Scrapy installed in your Python environment.
2. Create a Scrapy project (if not already created) and navigate to the project folder.
3. Create a new Scrapy spider by copying and pasting the provided Scrapy code into a Python file.
4. Adjust the start_urls to point to your desired source URLs (e.g., a CSV file containing the collected links).
5. Run the spider using the command: scrapy crawl LinkListsSpider -o links.csv (to collect links) and scrapy crawl MoviesSpider -o movies.json (to collect movie data).
6. The collected links and movie data will be saved in CSV and JSON formats, respectively.

Note: Ensure you have the required dependencies for each script installed, and customize the script according to your requirements before executing.