21st Feb Assignment

April 19, 2023

1 Assignment 20

Q1. What is Web Scraping? Why is it Used? Give three areas where Web Scraping is used to get data.

Ans. Web scraping is the automated process of extracting information from websites using software tools, also known as web crawlers or spiders. It involves sending requests to a website, parsing the HTML response, and extracting relevant data from the web pages.

Web scraping is used to gather large amounts of data from websites quickly and efficiently. It can be used for a variety of purposes, including market research, price comparison, content aggregation, and data analysis. Some common areas where web scraping is used to get data are:

- E-commerce: Web scraping is used to gather product information and pricing data from e-commerce sites. This data is used to monitor competitor prices, update product catalogs, and perform market research.
- Social media: Web scraping is used to gather data from social media platforms, including user profiles, posts, and comments. This data is used for sentiment analysis, trend analysis, and social listening.
- Finance: Web scraping is used to gather financial data, including stock prices, market trends, and economic indicators. This data is used for investment analysis, risk management, and financial modeling.

Q2. What are the different methods used for Web Scraping?

Ans. There are several methods and techniques used for web scraping, some of which are:

- Regular Expression (RegEx): Regular expressions are used to match patterns in the HTML code of a website. This method can be effective for scraping data from websites with a predictable structure, such as news websites or job boards.
- DOM Parsing: The Document Object Model (DOM) is a programming interface for HTML and XML documents. Web scraping tools can use DOM parsing to navigate through the document tree and extract specific elements from a web page.

- XPath: XPath is a query language for selecting nodes in an XML document. It can also be used for web scraping by selecting specific elements from a web page using XPath expressions.
- CSS Selectors: CSS selectors are used to target specific HTML elements based on their attributes, classes, or IDs. Web scraping tools can use CSS selectors to extract data from web pages by selecting specific elements based on their CSS classes or IDs.
- APIs: Some websites offer Application Programming Interfaces (APIs) that allow developers to access data in a structured format. APIs can be an effective method for web scraping if the website provides the data you need through an API.
- Headless Browsers: Headless browsers are web browsers that can be run in a headless mode, which means they can be automated to interact with web pages and extract data. This method can be effective for scraping data from websites that require JavaScript to be executed.

These methods can be used alone or in combination with each other, depending on the website's structure and the data that needs to be scraped.

Q3. What is Beautiful Soup? Why is it used?

Ans. Beautiful Soup is a Python library used for web scraping purposes. It is a powerful and easy-to-use tool that can help you parse HTML and XML documents to extract specific data.

Beautiful Soup is used to navigate through the HTML tree structure of a web page and extract data from specific HTML elements. It provides a simple and flexible API that allows you to search for HTML tags and attributes and extract the data contained within them.

Some of the features of Beautiful Soup include:

- Navigating and searching through the HTML tree structure using CSS selectors, tag names, and attributes.
- Parsing HTML and XML documents and extracting data from them.
- Handling invalid markup and cleaning up messy HTML.
- Converting parsed HTML documents into a more accessible data structure, such as a list or dictionary. #### Beautiful Soup is commonly used in web scraping projects because of its ease of use and flexibility. It allows you to quickly and easily extract data from websites without having to write complex code or deal with low-level details of HTML parsing. It also provides robust error handling and is able to handle malformed HTML, making it a reliable tool for web scraping.

Q4. Why is flask used in this Web Scraping project?

Ans. Flask is a lightweight web framework for Python that is commonly used to build web applications. Flask is often used in web scraping projects because it provides a simple way to create a web interface for your web scraping tool.

By using Flask in a web scraping project, you can create a web page that allows users to input search queries or specify parameters for the data they want to scrape. The Flask web application can then use your web scraping tool to extract the data from the web and display the results on the web page.

Flask is also easy to use and has a small footprint, making it a good choice for smaller web scraping projects that don't require the full features of a more complex web framework. Additionally, Flask has a large community of developers and extensive documentation, making it easy to find help and resources when building your web scraping application.

In summary, Flask is used in web scraping projects to provide a user-friendly interface for your web scraping tool and to make it easier to display and share your scraped data on the web.

Q5. Write the names of AWS services used in this project. Also, explain the use of each service.

Ans. Some of the AWS services are:

1. Amazon EC2 (Elastic Compute Cloud): EC2 is a virtual machine service that allows you to spin up and manage virtual machines on the cloud. In this project, you could use EC2 to host your web scraping script and run it on a virtual machine in the cloud.
2. Amazon S3 (Simple Storage Service): S3 is a cloud storage service that provides highly scalable and durable storage for your data. In this project, you could use S3 to store the data that you scrape from the websites.
3. Amazon DynamoDB: DynamoDB is a NoSQL database service that provides fast and flexible document and key-value data storage. In this project, you could use DynamoDB to store and query the data that you scrape from the websites.
4. AWS Lambda: Lambda is a serverless compute service that allows you to run your code in response to events and triggers. In this project, you could use Lambda to trigger your web scraping script periodically and then process the scraped data and store it in S3 or DynamoDB.
5. Amazon CloudWatch: CloudWatch is a monitoring service that provides metrics and logs for your AWS resources. In this project, you could use CloudWatch to monitor your EC2 instances, Lambda functions, and other AWS resources to ensure that they are running properly and to diagnose any issues that may arise.

These are just a few examples of how AWS services can be used in a web scraping project. Depending on the specific requirements and needs of your project, you may choose to use different AWS services or use them in different ways.

[ ]: