ELIEZER CARVALHO - SEPTEMBER 2025

Web Scraping

What is Web Scraping?

Web scraping is a technique used to **extract information from websites in an automated manner**. In this technique, a programme or script accesses web pages and searches for specific data on them. The script collects information such as text, images, links, or tables.

It is a useful tool in various areas, such as collecting prices for comparison, extracting articles for content analysis, or monitoring changes in offer or service pages.

The process begins when the script sends an **HTTP request** to the server to load the page content. Once loaded, the script analyses the **HTML** or **XML** of the page, searching for the desired data. It then stores it in files such as **CSV**, **Excel**, or **databases**.

API

APIs (Application Programming Interfaces) are the bridge between software systems. They enable different applications to communicate in a structured and efficient manner. They allow a **front-end (such as a website or app)** to access data or functionality from a **back-end (server)** without needing to understand the internal details of the system.

Communication between the front-end and back-end via **API** is done through HTTP calls, such as **GET** (to retrieve data), **POST** (to send data), **PUT** (to update) and **DELETE** (to delete). The front-end makes a request to the API, which searches for data in the back-end and returns it in a structured format such as JSON or XML, facilitating the processing and display of information.

This approach allows platforms to connect and exchange data in a simple manner, without exposing the internal complexity of the systems.

Web Scraping vs API: Which to Use?

Web scraping is the ideal option thanks to its extreme flexibility. Scraping is undoubtedly the best option when a website **does not** offer an API or when you need to access data that is not available in a public API. Websites such as e-commerce sites, marketplaces, or news pages often do not have APIs that expose all relevant data. **That's where scraping comes in!**

However, this flexibility also brings challenges. Scraping depends on the HTML structure of a page, which makes it more fragile and susceptible to breakdowns. If the website changes its layout or the HTML is altered, the scraping script may stop working and adjustments will need to be made. In addition, scraping is more time-consuming than accessing via API, especially when it is necessary to go through multiple pages or collect large volumes of data.

On the other hand, **APIs are faster and more structured**. These tools provide data directly in the desired format, often with filters and parameters that make data collection much more efficient. Instead of having to process an entire page of a weather forecast website to extract temperature data, you can simply make a request to the API to obtain the specific information in a well-organised format.

APIs are more stable. Data is delivered in a standardised format. This ensures that the code will not break when the website layout changes. In addition, many APIs offer request rate limits, which helps prevent server overload and ensures more responsible data collection.

It is important to note that **not all websites offer APIs**. Furthermore, even when they do exist, they often do not provide all relevant data. Many APIs require authentication and the provision of an API key, which is an additional hurdle. Request limits, i.e. the number of calls allowed per day, are a reality. **This means that there is a limit to the amount of data that can be obtained in a given period.**

Technologies and Libraries for Web Scraping

Selenium is a powerful library used for browser automation. It is especially useful for scraping dynamic websites that load content via **JavaScript**.

Originally developed for web application testing, it has become essential for data collection, allowing you to control browsers such as **Chrome**, **Firefox**, and **Safari**.

With Selenium, you can simulate interactions such as clicks, scrolling, and form filling, which is crucial for websites with dynamic content.

Playwright is the latest browser automation library. Unlike Selenium, it uses **asynchronous execution**, which significantly improves performance.

This software supports **Chromium**, **Firefox**, and **WebKit**, allowing for simple cross-browser testing. In contrast, Selenium requires extra configurations for different browsers.

Both handle dynamic websites well and allow interactions with the page, but **Playwright clearly stands out for its modern, faster, and easier-to-use API**. Selenium is more traditional, but **Playwright is the more efficient and flexible alternative for browser automation**.

Requests is undoubtedly the simplest and most popular library for making HTTP requests in Python. It is essential for obtaining content from web pages. Simplicity and intuitiveness are undoubtedly its main features.

This allows you to send **HTTP** requests with just a few lines of code. It is the ideal choice for scraping websites with static content, where the information is already loaded on the page, making the process faster and more straightforward.

Requests makes it easy to work with authentication and cookies, allowing you to handle sessions and basic or advanced authentication. This is useful when access to certain pages requires a login or other checks.

The ability to intercept API calls is another important feature. Requests allows you to make **requests to RESTful APIs** or interact with endpoints to obtain data more efficiently.

Requests is a powerful tool for scraping simple websites or working with APIs. It is not suitable for pages that rely on JavaScript to load content.

BeautifulSoup is an HTML and XML parsing library. It is often used in conjunction with **Requests** to facilitate the extraction of structured data from web pages. The API is simple and intuitive, allowing developers to easily navigate the page structure and extract the desired information, such as tags, attributes, and content from specific elements.

It is the ideal tool for working with HTML or XML documents, making it easier to read and manipulate these documents.

However, it is slower on large pages and does not handle content loaded via JavaScript, which limits its use on dynamic websites.

Pandas is not a tool designed for scraping, but it is extremely useful after data collection. It is essential for analysing, cleaning and manipulating the extracted data.

It allows you to organise and process data efficiently using ***DataFrames***, a structure that facilitates data analysis and transformation. This system allows you to read and write data in various formats, such as **CSV**, **Excel** and **SQL**.

A powerful tool for cleaning and transforming data that allows you to perform tasks such as removing duplicates, converting types, and aggregating data.

This facilitates the preparation of data for further analysis or visualisation.

Conclusion

Web scraping is undoubtedly a powerful technique that allows you to extract large volumes of data from the web. It offers a wide range of applications in different areas. The tools and libraries we have discussed, namely **Selenium**, **Playwright**, **Requests**, **BeautifulSoup**, and **Pandas**, are just some of the options that facilitate this process. Each has its own specific characteristics and functionalities to deal with different types of websites and scraping requirements.

The impact of web scraping is undeniable. It is not just about extracting data, but an essential tool in various fields, such as data analysis, **artificial intelligence** (AI), and **machine learning** (ML). Collecting structured data from the web allows you to train machine learning models, create predictive algorithms, and perform trend analysis on markets and consumer behaviour.

By combining the power of data extraction with analysis tools such as **Pandas**, you can transform raw data into valuable insights that guide business decisions, optimise operations, and drive technological innovation. However, it is crucial to respect ethical standards when scraping, such as usage limitations, privacy policies, and website terms of service, to ensure that data collection is done responsibly and legally.

Web scraping is more than just a collection tool — it is a key to unlocking the potential of valuable data on the internet, fuelling innovative technologies and empowering professionals in various fields to make more informed and accurate decisions.

"Data is the new oil. It’s valuable, but if unrefined it cannot really be used. It has to be changed into gas, plastic, chemicals, etc to create a valuable entity that drives profitable activity; so must data be broken down, analyzed for it to have value."

Clive Humby, 2006