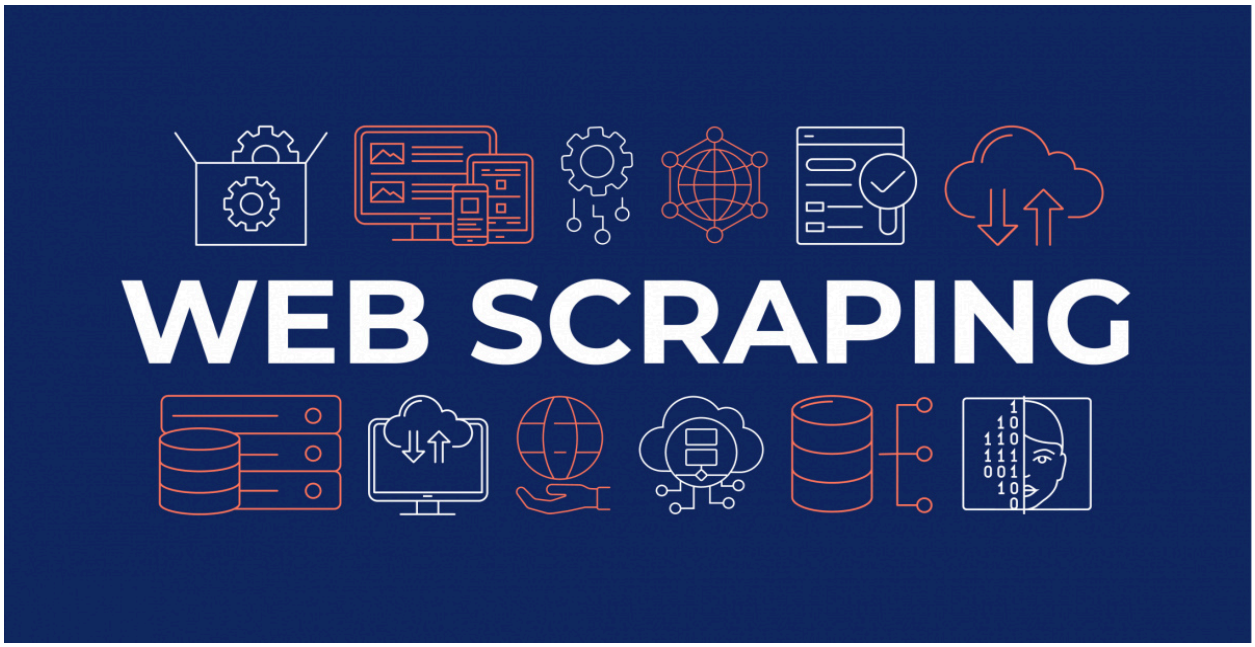
ELIEZER CARVALHO - SETEMBRO 2025

Web Scraping



O que é Web Scraping ?

O **web scraping** é uma técnica utilizada para **extrair informações de sites da internet de forma automatizada**. Nesta técnica, um programa ou script acede a páginas da web e procura dados específicos nas mesmas. O script recolhe informações como textos, imagens, links ou tabelas.

É uma ferramenta útil em várias áreas, como a recolha de preços para comparação, a extração de artigos para análise de conteúdo ou o monitoramento de mudanças em páginas de ofertas ou serviços.

O processo começa quando o script envia uma **solicitação HTTP** ao servidor para carregar o conteúdo da página. Uma vez carregada, o script analisa o **HTML** ou **XML** da página, procurando os dados desejados. Depois, armazena-os em arquivos como **CSV**, **Excel** ou **bancos de dados**.

API

As **APIs (Interface de Programação de Aplicações)** são a ponte entre sistemas de software. Permitem que diferentes aplicações se comuniquem de maneira estruturada e eficiente. Estas possibilitam que um **front-end (como um site ou app)** aceda a dados ou funcionalidades de um **back-end (servidor)**, sem necessidade de compreender os detalhes internos do sistema.

A comunicação entre front-end e back-end via **API** é feita através de chamadas HTTP, como **GET** (para buscar dados), **POST** (para enviar dados), **PUT** (para atualizar) e **DELETE** (para excluir). O front-end faz uma solicitação à API, que pesquisa os dados no back-end e os retorna num formato estruturado como JSON ou XML, facilitando o processamento e a exibição das informações.

Esta abordagem permite que as plataformas se conectem e troquem dados de maneira simples, sem expor a complexidade interna dos sistemas.

Web Scraping vs API: Qual Usar?

O **Web Scraping** é a opção ideal, graças à sua extrema flexibilidade. O scraping é, sem dúvida, a melhor opção quando um site **não** oferece uma API ou quando é necessário aceder a dados que não estão disponíveis numa API pública. Sites como e-commerce, marketplaces ou páginas de notícias frequentemente não têm APIs que expõem todos os dados relevantes. **É aí que o scraping entra!**

No entanto, essa flexibilidade também traz desafios. O scraping depende da estrutura do HTML de uma página, o que o torna mais frágil e suscetível a quebras. Se o site mudar de layout ou o HTML for alterado, o script de scraping pode deixar de funcionar e será necessário fazer ajustes. Além disso, o scraping é mais demorado do que o acesso via API, especialmente quando é necessário percorrer várias páginas ou recolher grandes volumes de dados.

Por outro lado, as **APIs são mais rápidas e estruturadas**. Estas ferramentas fornecem os dados diretamente no formato pretendido, frequentemente com filtros e parâmetros que tornam a recolha de dados muito mais eficiente. Em vez de ter de processar uma página inteira de um site de previsão do tempo para extrair dados sobre a temperatura, basta fazer uma requisição à API para obter a informação específica num formato bem organizado.

As APIs são mais estáveis. Os dados são entregues num formato padronizado. Isto garante que o código não quebrará quando o layout do site mudar. Além disso, muitas APIs oferecem taxas de limite de requisições, o que ajuda a evitar sobrecarregar os servidores e garante uma recolha de dados mais responsável.

É importante ter em conta que nem **todos os sites oferecem APIs**. Além disso, mesmo quando elas existem, frequentemente não disponibilizam todos os dados relevantes. Muitas APIs exigem autenticação e o fornecimento de uma chave de API, o que constitui um obstáculo adicional. Os limites de requisição, ou seja, a quantidade de chamadas permitidas por dia, são uma realidade. **Isto significa que há um limite à quantidade de dados que se pode obter num determinado período.**

Tecnologias e Bibliotecas para Web Scraping

Selenium é uma biblioteca poderosa e usada para automação de navegadores. É especialmente útil para scraping de sites dinâmicos que carregam conteúdo via **JavaScript**.

Originalmente desenvolvido para testes de aplicações web, tornou-se essencial para a recolha de dados, permitindo controlar navegadores como o **Chrome**, o **Firefox** e o **Safari**.

Com Selenium, é possível simular interações como cliques, *scroll* e preenchimento de formulários, sendo crucial para sites com conteúdo dinâmico.

Playwright é a mais recente biblioteca de automação de navegadores. Ao contrário do Selenium, utiliza **execução assíncrona**, o que melhora significativamente o desempenho.

Este software suporta **Chromium**, **Firefox** e **WebKit**, permitindo testes cross-browser de forma simples. Em contraste, o Selenium requer configurações extras para diferentes navegadores.

Ambos lidam bem com sites dinâmicos e permitem interações com a página, mas o **Playwright destaca-se claramente pela sua API moderna, mais rápida e de fácil uso**. O Selenium é mais tradicional, **mas o Playwright é a alternativa mais eficiente e flexível para a automação de navegadores**.

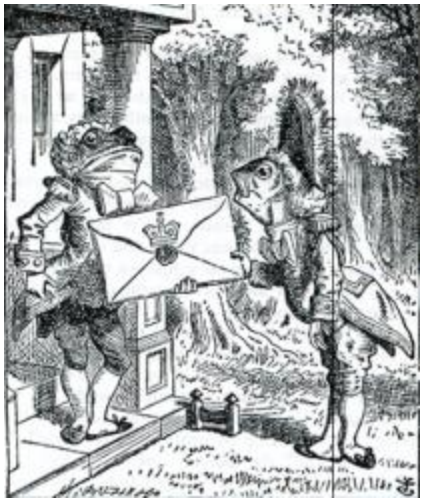
Requests é, sem dúvida, a biblioteca mais simples e popular para fazer requisições HTTP em Python. É fundamental para obter o conteúdo de páginas web. A simplicidade e a intuitividade são, sem dúvida, a sua principal característica.

Isto permite enviar requisições **HTTP** com poucas linhas de código. É a escolha ideal para scraping de sites com conteúdo estático, onde as informações já estão carregadas na página, tornando o processo mais rápido e direto.

Requests facilita o trabalho com autenticação e cookies, permitindo lidar com sessões e autenticação básica ou avançada. Isto é útil quando o acesso a determinadas páginas exige login ou outras verificações.

A capacidade de interceptar chamadas de API é outra funcionalidade importante. Requests permite realizar **requisições a APIs RESTful** ou interagir com endpoints para obter dados de forma mais eficiente.

Requests é uma ferramenta poderosa para fazer scraping em sites simples ou para trabalhar com APIs. Não é adequado para páginas que dependem de JavaScript para carregar conteúdo.



BeautifulSoup é uma biblioteca de parsing de HTML e XML. É frequentemente usada em conjunto com o **Requests** para facilitar a extração de dados estruturados de páginas da web. A API é simples e intuitiva, permitindo que os desenvolvedores naveguem facilmente pela estrutura da página e extraiam as informações desejadas, como tags, atributos e conteúdo de elementos específicos.

É a ferramenta ideal para trabalhar com documentos HTML ou XML, facilitando a leitura e a manipulação desses documentos.

No entanto, é mais lenta em páginas grandes e não lida com conteúdo carregado via JavaScript, o que limita o seu uso em sites dinâmicos.

Pandas não é uma ferramenta voltada para scraping, mas é extremamente útil após a recolha de dados. É essencial para análise, limpeza e manipulação dos dados extraídos.

Permite organizar e tratar os dados de forma eficiente, recorrendo a ***DataFrames***, uma estrutura que facilita a análise e a transformação dos dados. Este sistema permite ler e escrever dados em diversos formatos, como **CSV**, **Excel** e **SQL**.

Uma poderosa ferramenta para limpeza e transformação de dados que permite realizar tarefas como remoção de duplicados, conversões de tipos e agregações de dados.

Isto facilita a preparação dos dados para análise ou visualização posterior.

Conclusão

O Web Scraping é, sem dúvida, uma técnica poderosa que permite extrair grandes volumes de dados da web. Oferece uma vasta gama de aplicações em diferentes áreas. As ferramentas e bibliotecas que discutimos, nomeadamente o **Selenium**, o **Playwright**, o **Requests**, o **BeautifulSoup** e o **Pandas**, são apenas algumas das opções que facilitam esse processo. Cada uma tem as suas características e funcionalidades específicas para lidar com diferentes tipos de sites e requisitos de scraping.

O impacto do Web Scraping é inegável. Não se trata apenas de extrair dados, mas de uma ferramenta essencial em diversos campos, como a análise de dados, a **inteligência artificial** (IA) e ***Machine Learning*** (ML). A recolha de dados estruturados da web permite treinar modelos de machine learning, criar algoritmos preditivos e realizar análises de tendências em mercados e comportamentos de consumidores.

Ao combinar o poder de extração de dados com ferramentas de análise como o **Pandas**, é possível transformar dados brutos em insights valiosos que orientam decisões de negócios, otimizam operações e impulsionam inovações tecnológicas. No entanto, é crucial respeitar normas éticas ao realizar scraping, como limitações de uso, políticas de privacidade e termos de serviço dos sites, para garantir que a recolha de dados seja feita de forma responsável e legal.

O Web Scraping é mais do que uma simples ferramenta de recolha— é uma chave para desbloquear o potencial de dados valiosos da internet, alimentando tecnologias inovadoras e capacitando profissionais de várias áreas a tomar decisões mais informadas e precisas.

"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