Comparativo de Bibliotecas para Web Scraping em Python

Allyson de Almeida Sirvano

19 de maio de 2025

---

1. Introdução

O web scraping é uma técnica amplamente utilizada para extração automatizada de dados a partir de páginas da web. Na linguagem Python, diversas bibliotecas estão disponíveis para essa finalidade, cada uma com suas vantagens e aplicações específicas. Este documento apresenta uma análise comparativa entre três das principais bibliotecas utilizadas: **Beautiful Soup**, **Selenium** e **Playwright**.

2. Beautiful Soup

2.1. Descrição

Beautiful Soup é uma biblioteca Python projetada para realizar parsing de documentos HTML e XML. É amplamente utilizada em conjunto com bibliotecas como `requests`, que realizam as requisições HTTP. É ideal para páginas web com conteúdo estático.

Características:

- Ideal para páginas estáticas e bem estruturadas.
- Não renderiza JavaScript.
- Leve, simples e de rápida implementação.

2.2. Exemplo de Uso

Instalação:

```
pip install beautifulsoup4 requests
```

Código:

```
import requests
from bs4 import BeautifulSoup

url = "https://exemplo.com"
response = requests.get(url)
soup = BeautifulSoup(response.text, "html.parser")
```

```
titulos = soup.find_all("h1")
for t in titulos:
    print(t.text)
```

3. Selenium

3.1. Descrição

Selenium é uma ferramenta que permite a automação de navegadores reais. É útil para realizar scraping em páginas dinâmicas, que carregam conteúdo via JavaScript. Também permite simular cliques, rolagem, preenchimento de formulários e outras ações do usuário.

Características:

- Ideal para páginas com conteúdo dinâmico.
- Requer driver do navegador (como o ChromeDriver).
- Suporta interações complexas com a página.

3.2. Exemplo de Uso

Instalação:

```
pip install selenium
```

Código:

```
from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()
driver.get("https://exemplo.com")

titulos = driver.find_elements(By.TAG_NAME, "h1")
for t in titulos:
    print(t.text)

driver.quit()
```

4. Playwright

4.1. Descrição

Playwright é uma biblioteca moderna para automação de navegadores, desenvolvida pela Microsoft. É uma alternativa robusta ao Selenium, com melhor desempenho e suporte nativo a múltiplos navegadores. Permite controle total sobre o contexto de navegação, cookies, autenticação, entre outros.

Características:

- Ideal para páginas modernas com alto uso de JavaScript.
- Alta performance e controle de rede.
- Suporte a múltiplos contextos de navegador.

4.2. Exemplo de Uso

Instalação:

```
pip install playwright
playwright install
```

Código:

```
from playwright.sync_api import sync_playwright

with sync_playwright() as p:
    browser = p.chromium.launch()
    page = browser.new_page()
    page.goto("https://exemplo.com")

    titulos = page.query_selector_all("h1")
    for t in titulos:
        print(t.inner_text())

    browser.close()
```

5. Comparativo Geral

Critério	Beautiful Soup	Selenium	Playwright
Suporte a JavaScript	Não	Sim	Sim
Velocidade	Alta	Baixa	Alta
Complexidade	Baixa	Média	Média/Alta
Interação com página	Não	Sim	Sim
Controle de rede	Não	Não	Sim
Melhor uso para	HTML estático	Navegação real	JS avançado

Tabela 1: Tabela comparativa entre as bibliotecas de Web Scraping

Cada biblioteca possui seu campo de aplicação ideal. O Beautiful Soup é extremamente eficaz para páginas estáticas e simples. Já o Selenium e o Playwright são mais adequados para conteúdos dinâmicos, com o Playwright se destacando por performance e controle. A escolha da ferramenta depende do tipo de página a ser acessada, da complexidade do scraping e da necessidade de interações com o conteúdo.