Web Scraping com I.A.

Construindo extratores de dados resilientes*

Felipe Guilherme Sabino

** mas não invencíveis!* 😄

Sobre mim

- **Nome:** Felipe Guilherme Sabino
- **Idade:** 33 anos
- **Família:** Casado com Evelyn, filha Laura (4 anos)
- **Naturalidade:** Joinville
- **Experiência:** +15 anos em desenvolvimento
- **Empresas:** TOTVS, Conta Azul, Nubank, Transfeera
- **Empreendimento:** Judite.App - IA para juristas

Ainda sobre mim

- **Paixão:** Fazer coisas aparentemente impossíveis
- **Valorização:** Jornada > Final
- **Hobbies:** Colecionador de video games antigos
- **Interesse:** Reproduzir mecânicas de jogos

Agora sim!

Web Scraping com I.A.

Construindo extratores de dados resilientes*

Mas o que é Web Scraping?

Senta que lá vem a história...

Projeto na Conta Azul

Objetivo: Construir base de CNPJs

Desafio: Automatização de captcha

Ferramenta usada: Selenium

Exemplo de Código: Selenium

```
from selenium import webdriver

driver = webdriver.Chrome()
driver.get('http://example.com')
element = driver.find_element_by_id('some-id')
print(element.text)
driver.quit()
```

Scrapy

```
import scrapy

class QuotesSpider(scrapy.Spider):
    name = "quotes"
    start_urls = [
        'http://quotes.toscrape.com/page/1/',
    ]
    def parse(self, response):
        for quote in response.css('div.quote'):
            yield {
                'text': quote.css('span.text::text').get(),
                'author': quote.css('span small::text').get(),
                'tags': quote.css('div.tags a.tag::text').getall(),
            }
        next_page = response.css('li.next a::attr(href)').get()
        if next_page is not None:
            yield response.follow(next_page, self.parse)
```

Nightmare.js

```
const Nightmare = require('nightmare')
const nightmare = Nightmare({ show: true })

nightmare
  .goto('https://duckduckgo.com')
  .type('#search_form_input_homepage', 'github nightmare')
  .click('#search_button_homepage')
  .wait('#r1-0 a.result__a')
  .evaluate(() => document.querySelector('#r1-0 a.result__a').href)
  .end()
  .then(console.log)
  .catch(error => {
    console.error('Search failed:', error)
  })
```

lxml

```
from lxml import html
import requests

response = requests.get('http://example.com')
tree = html.fromstring(response.content)
title = tree.xpath('//title/text()')
print(title)
```

aiohttp

```python
import aiohttp
import asyncio

async def fetch(url):
    async with aiohttp.ClientSession() as session:
        async with session.get(url) as response:
            return await response.text()

async def main():
    html = await fetch('http://example.com')
    print(html)

asyncio.run(main())
```

Puppeteer

```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('http://example.com');
  const title = await page.title();
  console.log(title);
  await browser.close();
})();
```

Casos de Uso

Transfeera

Utilizava "robôs" para automatizar backoffice

https://github.com/negherbon/braziljs-bb-crawler

Money Advisor

Bot para organizar finanças usando scraping

Pitch no Startup Weekend (2016)

Guia Bolso

- Adquirido pelo PicPay
- Buscava dados de contas bancárias usando scraping

Olivia App

- Adquirido pelo Nubank
- Buscava dados das contas bancárias

Scraping com LLM

Firecrawl e Jina AI

- **Firecrawl:** Scraping usando LLM
- **Jina AI Reader:** Facilita o scraping

Mão na massa!

- Configuração do ambiente
- Execução de scripts de scraping
- Utilização de LLM para extração de dados

Ferramentas Necessárias

- **VSCode**
- **Git**
- **API Keys**

Passos para Configuração

1. **Instalar VSCode**
2. **Clonar Repositório**
3. **Configurar Ambiente Python**
4. **Adicionar API Keys**

Recursos Adicionais

- Firecrawl
- Jina AI Reader

Perguntas?

- Fique à vontade para perguntar!

- **Resumo:** Web Scraping com LLM
- **Contato:** sabino@judite.app
- **LinkedIn:** linkedin.com/in/fgsabino
- **GitHub:** github.com/sabino
- **Instagram:** instagram.com/sabino
- **Twitter:** x.com/fgsabino

Obrigado!