

Machine Learning Engineer

Python para ML e IA

Desenvolvimento de API com FastAPI

Leonardo Pena

/ Seja muito bem vindo



OBJETIVO

Selenium para scraping, rotas protegidas que extraem dados de páginas



COMPARATIVO

Veremos como rodar Chrome em modo headless com webdriver



EXEMPLO

Criaremos funções `get_title`, `get_content`, `get_links_and_images`

Objetivo dessa primeira parte



Passo 1

Explicar
instalação e
configuração
do Selenium e
ChromeDriver



Passo 2

Implementar
`init_driver()`
para modo
headless



Passo
3

Mostrar
funções de
scraping e
como tratar
exceções



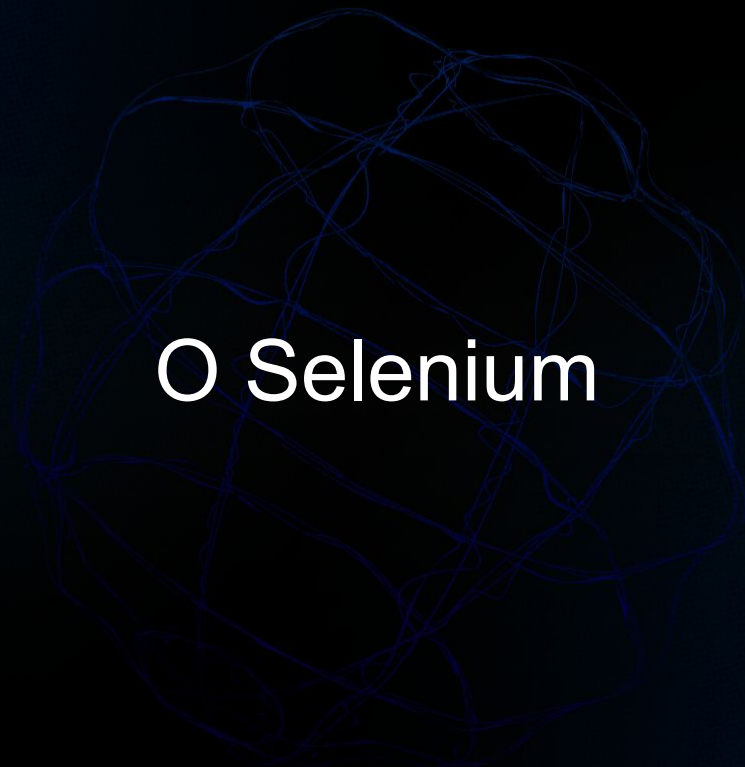
Passo 4

Criar rotas
GET
protegidas
com
`Depends(verify_password)`



Passo
5

Demonstrar
doc
automática e
testes
práticos



O Selenium



/ O que é?



Selenium

- Ferramenta poderosa para automação de navegadores web
- Suporta múltiplas linguagens de programação como Python, Java e C#
- Compatível com diversos navegadores, incluindo Chrome, Firefox e Safari
- Permite a criação de testes automatizados para interfaces de usuário
- Facilita integração contínua e pipelines de CI/CD

/ O que é?



Selenium

- O Selenium é composto por vários componentes, sendo os mais utilizados:
 - **Selenium WebDriver:** Interface para interagir com diferentes navegadores de forma automatizada.
 - **Selenium IDE:** Ferramenta para gravar e reproduzir interações com o navegador.
 - **Selenium Grid:** Permite executar testes em múltiplos ambientes e navegadores simultaneamente.

/ Configuração do WebDriver



Selenium

O Selenium precisa de um WebDriver específico para o navegador que você deseja automatizar. Por exemplo, para o Google Chrome, você precisará do ChromeDriver.

/ No nosso exemplo
usaremos o Google
Colab



Selenium



- **Instalação das Dependências**

- Para utilizar o Selenium no Google Colab, precisamos instalar algumas dependências, incluindo o navegador Google Chrome e o ChromeDriver compatível.

Passo 1: Instalar o Google Chrome

Primeiro, vamos instalar o Google Chrome no ambiente do Colab.



O código gerado pode estar sujeito a uma licença |

```
# Atualizar a lista de pacotes
```

```
!apt-get update
```

```
# Instalar o Google Chrome
```

```
!apt-get install -y wget unzip
```

```
!wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
```

```
!dpkg -i google-chrome-stable_current_amd64.deb
```

```
!apt-get -f install -y
```

<https://colab.research.google.com/drive/1ZFR0g2VcCyyvFzGFeyx1Kbbjw0wPVkq2?usp=sharing>

Passo 2: Instalar o ChromeDriver

Em seguida, instalaremos o ChromeDriver que é compatível com a versão do Google Chrome instalada.

```
[12] # Verificar a versão do Google Chrome instalada  
!google-chrome --version
```

```
Google Chrome 131.0.6778.264
```

```
# Instalar o webdriver-manager  
!pip install webdriver-manager
```

<https://colab.research.google.com/drive/1ZFR0g2VcCyyvFzGFeyx1Kbbjw0wPVkq2?usp=sharing>

Passo 3: Importação das Bibliotecas Necessárias



```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from webdriver_manager.chrome import ChromeDriverManager
import time
```

<https://colab.research.google.com/drive/1ZFR0g2VcCyyvFzGFeyx1Kbbjw0wPVkq2?usp=sharing>

Passo 4: Configuração das Opções do Chrome

```
# Configurar opções do Chrome
chrome_options = Options()
chrome_options.add_argument("--headless") # Executar no modo headless (sem interface gráfica)
chrome_options.add_argument("--no-sandbox")
chrome_options.add_argument(["--disable-dev-shm-usage"])
```

<https://colab.research.google.com/drive/1ZFR0g2VcCyyvFzGFeyx1Kbbjw0wPVkq2?usp=sharing>

Passo 5: Inicialização do WebDriver com WebDriver Manager



```
# Inicializar o WebDriver com WebDriver Manager
service = Service(ChromeDriverManager().install())
driver = webdriver.Chrome(service=service, options=chrome_options)
```

<https://colab.research.google.com/drive/1ZFR0g2VcCyyvFzGFeyx1Kbbjw0wPVkq2?usp=sharing>

Passo 6: Automatizando a Pesquisa no Google

```
# Abrir o Google
driver.get('https://www.google.com')

# Encontrar a barra de pesquisa
search_box = driver.find_element(By.NAME, 'q')

# Digitar a consulta e enviar
search_query = 'Pos Tech FIAP'
search_box.send_keys(search_query)
search_box.submit()

# Aguardar os resultados carregarem
time.sleep(3)

# Capturar os títulos dos resultados
titles = driver.find_elements(By.TAG_NAME, 'h3')
print(f"Resultados para a pesquisa: '{search_query}'\n")
for idx, title in enumerate(titles, start=1):
    print(f"{idx}. {title.text}")

# Fechar o navegador
driver.quit()
```

⇒ Resultados para a pesquisa: 'Pos Tech FIAP'

1. FIAP: Pós Tech – Pós-Graduações Tech Intensive nas áreas ...
2. Data Analytics
3. Software Architecture
4. Machine Learning ...
5. Defensive Cyber Security
6. Full Stack Development PÓS ...
7. MBA
8. FIAP Pós Tech – Software Architecture – Tech Challenge
9. Pos Tech Fiap + Alura
10. Pós Tech FIAP Alura
11. fbiopereira/movies-api-pos-tech-fiap
12. Pós Graduação Tech FIAP, vale a pena? : r/brdev

POSTECH

FIAP + alura