



RPA, **na** PRÁTICA

Aula – Selenium – Navegador

Instalação da biblioteca

Para instalar a lib, será usado o gerenciador de pacotes pip com o seguinte comando:

```
pip install selenium
```

Importar a biblioteca

Para importar o mínimo do Selenium, o comando será:

```
from selenium import webdriver
```

Abrir o Webdriver

Na nova versão do Selenium, o "**executable_path**" deixou de existir e precisa do **Service** para que o mesmo consiga abrir o navegador.

Para abrí-lo no novo formato, deverá ser feito o seguinte código:

```
from selenium.webdriver.chrome.service import Service

s_dir = Service('chromedriver.exe')
driver = webdriver.Chrome(service=s_dir)
```

Já a importação no formato antigo é realizado da seguinte maneira:

```
driver = webdriver.Chrome('chromedriver.exe')
```



Acessar um site

Com o "webdriver" salvo dentro da variável "driver", a mesma carrega muitas funcionalidades do navegador instanciado.

O comando primordial é para acessar um site:

```
driver.get('https://www.amazon.com.br/')
```

Maximizar a janela do navegador

O navegador também possui a capacidade de ser minimizado ou maximizado e no Selenium os comandos são:

```
driver.minimize_window()
```

ou

```
driver.maximize_window()
```

Colocar um tempo para carregamento – IMPLÍCITO

O carregamento implícito se aplica para todo e qualquer elemento na página. Então depois de instanciado uma vez, sempre que buscar alguma coisa na página, ele terá esse tempo máximo de espera para localizar o objeto.

A sintaxe dele é:

```
driver.implicitly_wait(30)
```



Colocar um tempo para carregamento – EXPLÍCITO

Em casos onde apenas um elemento nos interessa, pode ser utilizado um carregamento explícito, isto é, no código será apontado exatamente qual o objeto que se deseja aguardar um tempo máximo até que ele apareça. Esse método não se aplica à página como um todo.

A sintaxe dele é:

```
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

WebDriverWait(driver, 30).until(EC.presence_of_element_located((By.ID,
'twotabsearchtextbox')))
```

Fechar aba do navegador

O navegador também permite abrir várias abas e caso seja necessário fechar uma delas, o comando a seguir encerrará aquela que está em uso:

```
driver.close()
```

Fechar navegador e seu webdriver

Para que a instância do navegador seja encerrada totalmente, o comando a seguir precisa estar ao final do script:

```
driver.quit()
```





RPA, **na** PRÁTICA