



t2m
test to market

IQS

Teste automatizado – Front end

O que veremos neste módulo?

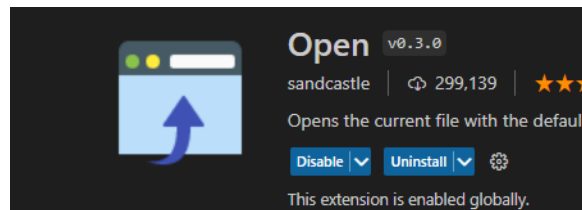
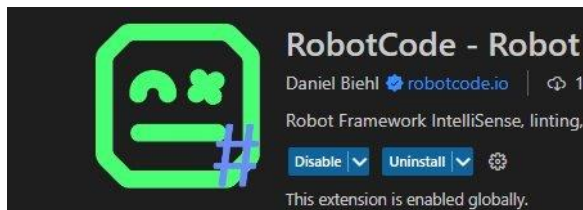
- Configurando o ambiente
- Explorando o Selenium
- Localizando Elementos
- Criando Validações

Configurando o ambiente

Verificando pré-requisitos de instalação

CA Prompt de Comando

```
Microsoft Windows [versão 10.0.19045.3930]  
(c) Microsoft Corporation. Todos os direitos reservados.  
  
C:\Users\ErculesMauriciodeSiq>python --version  
Python 3.12.1  
  
C:\Users\ErculesMauriciodeSiq>
```



Instalando libraries essenciais

É necessário instalar o Robot Framework com o Selenium via pip. Para isso, execute o seguinte comando:

```
pip install robotframework-seleniumlibrary==6.2.0
```

A screenshot of a Windows command prompt window. The title bar is not visible. The text 'C:\Users\ErculesMauricioDeSiqueira>' is displayed in a light blue font on a black background. A white mouse cursor is visible at the bottom right of the window.

```
C:\Users\ErculesMauricioDeSiqueira>
```

Dúvidas ...



Explorando o Selenium

Conhecendo as seções do robot

Settings

- a) Importar bibliotecas de teste.
- b) Definir metadados para test suites e test cases.

Variables

- a) Definir variáveis globais para todo o conjunto de testes.

Keywords

- a) Customizar keywords a partir de keywords existentes.

Test Cases

- a) Criar casos de teste a partir de keywords disponíveis.

```
*** Settings ***
```

```
*** Variables ***
```

```
*** Keywords ***
```

```
*** Test Cases ***
```

Começando...

Na seção 'Settings', importamos a biblioteca do Selenium, permitindo-nos utilizar suas funcionalidades para interagir com elementos da página web. Enquanto isso, na seção 'Test Cases', iniciamos uma sessão do navegador com a keyword 'Open Browser', preparando o terreno para a execução dos testes automatizados.

[SeleniumLibrary](#)

```
*** Settings ***  
Library      SeleniumLibrary  
  
*** Variables ***  
  
*** Keywords ***  
  
*** Test Cases ***  
TC001 - Realizar login com usuário válido  
    Open Browser    url=https://www.saucedemo.com/v1/    browser=chrome
```

Manipulando o navegador

Open Browser url=\${url} browser=\${brs}

A keyword "Open Browser" abre uma nova instância do navegador especificado, permitindo que a automação dos testes interaja com a página da web.

url: especifica o endereço da web (URL) da página que será aberta no navegador.

browser: define qual navegador será utilizado para abrir a página especificada. Este parâmetro permite que você escolha entre diferentes navegadores suportados pelo Selenium, como Chrome, Firefox, Safari, entre outros.

Maximize Browser Window

A keyword "Maximize Browser Window" maximiza a janela do navegador atual, ajustando-a para ocupar todo o espaço disponível na tela.

Close Browser

A keyword "Close Browser" fecha a janela do navegador atual que está sendo controlado pelo Selenium. Isso encerra a sessão do navegador aberta durante a execução dos testes automatizados.

Interagindo com o navegador

Click Element locator=\${lct}

A keyword "Click Element" é utilizada para simular o clique em um elemento específico da página web durante a automação de testes.

locator: especifica como encontrar o elemento na página. Ele pode ser definido usando diferentes estratégias de localização, como ID, classe CSS, XPath, etc.

Input Text locator=\${lct} text=\${txt}

A keyword "Input Text" é usada para inserir texto em um campo de entrada específico em uma página da web durante a automação de testes.

locator: especifica como encontrar o elemento na página. Ele pode ser definido usando diferentes estratégias de localização, como ID, classe CSS, XPath, etc.

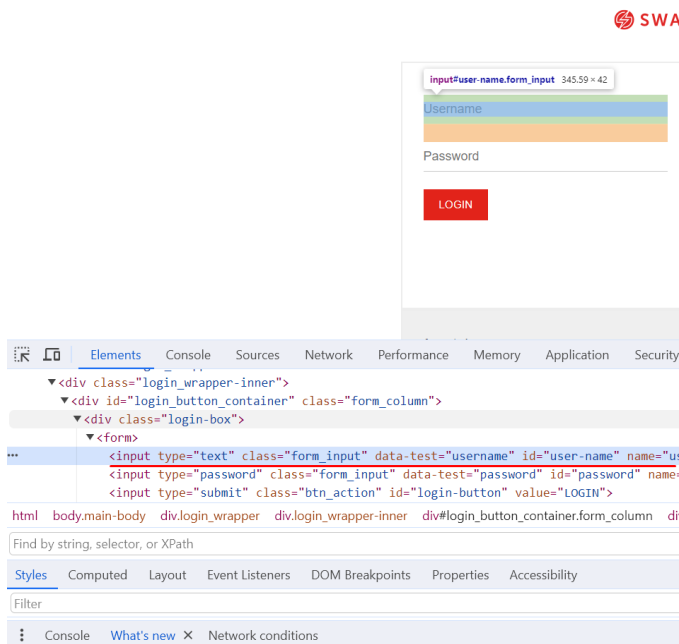
text: especifica o texto que será inserido no campo de entrada. Ele é utilizado para fornecer o conteúdo que será digitado na página da web durante a execução dos testes automatizados.

Dúvidas ...



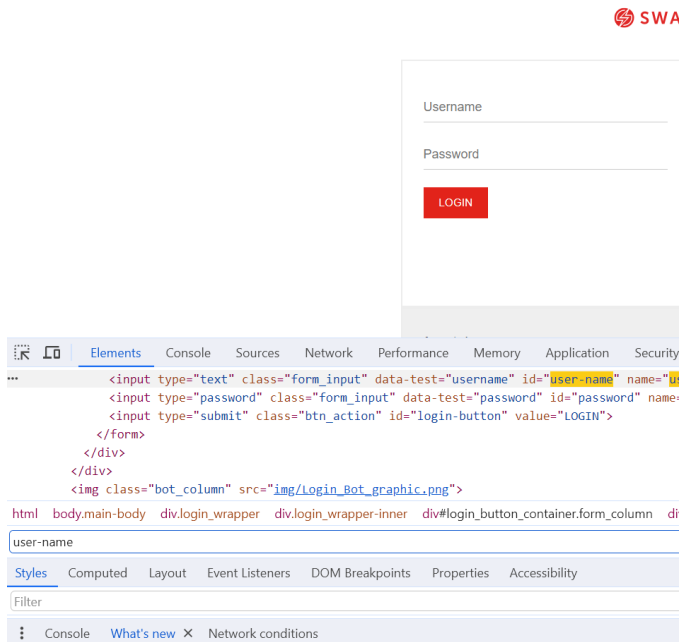
Localizando elementos

Encontrando as propriedades dos elementos



Para encontrar as propriedades dos elementos no navegador, basta clicar com o botão direito no elemento desejado e selecionar 'Inspecionar' para abrir o DevTools e visualizar as propriedades detalhadas do elemento na aba 'Elements'.

Encontrando as propriedades dos elementos



Dentro das ferramentas de desenvolvedor do Chrome, podemos filtrar elementos usando os seletores disponíveis, como ID, classe, nome e XPath. Isso nos permite localizar especificamente o elemento desejado na árvore DOM da página. Basta inserir o seletor desejado na barra de pesquisa.

Exemplos...

- **Id**

Para localizar um elemento pelo ID, procure pela propriedade 'id' nas propriedades do elemento.

Exemplo: user-name

- **Xpath**

Para localizar um elemento pelo XPath, utilize uma expressão XPath para identificar o elemento na árvore DOM da página.

Exemplo: `//input[@data-test='username']`

[Como localizar um elemento usando XPath](#)

- **Name**

Para localizar um elemento pelo atributo 'name', procure pela propriedade 'name' nas propriedades do elemento.

Exemplo: user-name

- **Css Selector**

Para localizar um elemento pelo seletor CSS, utilize uma expressão CSS que corresponda ao elemento desejado.

Exemplo: `input[placeholder='Username']`

[Como localizar um elemento usando Css Selector](#)

Dúvidas ...



Explorando o Selenium ₂

Page Object Model (POM)

O padrão Page Object Model é utilizado para organizar os elementos de uma página em seus próprios objetos, facilitando a manutenção do código e a reutilização dos elementos em diferentes partes do teste.

```
### Page Object Model (POM) ###

&{LOGIN_PAGE}
...   UsernameInput=id:user-name
...   PasswordInput=xpath://input[@name='password']
...   LoginButton=css:[class=btn_action]

*** Keywords ***

*** Test Cases ***
TC001 - Realizar login com usuário válido
    Open Browser    url=${URL}    browser=${BROWSER}
    Maximize Browser Window
    Input Text      ${LOGIN_PAGE.UsernameInput}    standard_us
    Input Text      ${LOGIN_PAGE.PasswordInput}    secret_sau
    Click Element   ${LOGIN_PAGE.LoginButton}
    Close Browser
```

Esperando por elementos interativos

Wait Until Element Is Visible locator=\${lct}

A keyword "Wait Until Element Is Visible" é usada para esperar até que um elemento específico na página da web esteja visível antes de continuar a execução do teste.

locator: especifica como encontrar o elemento na página. Ele pode ser definido usando diferentes estratégias de localização, como ID, classe CSS, XPath, etc.

Wait Until Element Is Enabled locator=\${lct}

A keyword "Wait Until Element Is Enabled" é utilizada para aguardar até que um elemento específico na página da web esteja habilitado para interação, ou seja, que esteja pronto para receber entrada do usuário, como cliques ou preenchimento de formulários.

locator: especifica como encontrar o elemento na página. Ele pode ser definido usando diferentes estratégias de localização, como ID, classe CSS, XPath, etc.

Trabalhando com keywords personalizadas

```
*** Keywords ***

### Ações ###

Realizar Login
    Input Text    ${LOGIN_PAGE.UsernameInput}    ${USUARIO_VA
    Input Text    ${LOGIN_PAGE.PasswordInput}    ${SENHA}
    Click Element    ${LOGIN_PAGE.LoginButton}

*** Test Cases ***

TC001 - Realizar Login com usuário válido
    Open Browser    url=${URL}    browser=${BROWSER}
    Maximize Browser Window
    Realizar login
    Close Browser
```

É fundamental usar keywords personalizadas para evitar a duplicação de código e promover a reutilização. Ao criar keywords para ações comuns ou repetitivas, podemos encapsular a lógica de automação em uma única função. Isso não apenas torna o código mais conciso e legível, mas também facilita a manutenção e atualização do teste.

Trabalhando com keywords personalizadas

```
*** Keywords ***
```

```
### Ações ###
```

```
Realizar login com ${username} e ${password}
```

```
Input Text    ${LOGIN_PAGE.UsernameInput}    ${username}
```

```
Input Text    ${LOGIN_PAGE.PasswordInput}    ${password}
```

```
Click Element  ${LOGIN_PAGE.LoginButton}
```

```
*** Test Cases ***
```

```
TC001 - Realizar login com usuário válido
```

```
Open Browser  url=${URL}    browser=${BROWSER}
```

```
Maximize Browser Window
```

```
Realizar login com ${USUARIO_VALIDO} e ${SENHA}
```

```
Close Browser
```

O uso de keywords com parâmetros é fundamental para tornar as automações mais flexíveis e reutilizáveis. Os parâmetros permitem que uma mesma keyword seja utilizada de maneira dinâmica em diferentes contextos, adaptando-se às necessidades específicas de cada cenário de teste.

Test Setup x Test Teardown

Test Setup

A keyword "Test Setup" é usada para configurar o ambiente de teste antes da execução dos casos de teste.

Test Teardown

A keyword "Tests Teardown" é usada para executar ações de limpeza ou finalização após a conclusão de cada teste em um conjunto de testes.

```
*** Settings ***  
  
Library    SeleniumLibrary  
  
Test Setup    Run Keywords  
...          Open Browser    url=${URL}    browser=${BROWSER}    AND  
...          Maximize Browser Window  
  
Test Teardown    Close Browser  
  
*** Variables ***  
*** Keywords ***  
*** Test Cases ***  
  
TC001 - Realizar Login com usuário válido  
    Realizar login com ${USUARIO_VALIDO} e ${SENHA}
```


Dúvidas ...



Criando validações

Obtendo informações da página

Get Text locator=\${lct}

A keyword "Get Text" é usada para extrair o texto de um elemento específico na página da web durante a automação de testes.

locator: especifica como encontrar o elemento na página. Ele pode ser definido usando diferentes estratégias de localização, como ID, classe CSS, XPath, etc.

Get Location

A keyword "Get Location" é usada para obter a URL da página atual durante a automação de testes

Validando conteúdo da página

Should Be Equal first=\${frt} second=\${snd}

A keyword "Should Be Equal" é usada para validar se dois valores são iguais durante a execução de testes automatizados. Ela compara dois valores e considera o teste bem-sucedido se eles forem iguais e falha se eles forem diferentes.

first: é o valor que você está esperando que seja igual a outro valor durante a execução do teste.

second: é o valor de referência que você está comparando com o primeiro valor durante a execução do teste.

Should Contain container=\${cnt} item=\${it}

A keyword "Should Contain" é usada para validar se uma determinada string está contida em outra string durante a execução de testes automatizados.

container: é a string principal ou o texto no qual estamos verificando se uma determinada string está contida.

item: é a string que estamos procurando dentro do "container", que é a string principal na qual estamos buscando a presença da outra string.

Dúvidas ...



Parabéns pela conclusão de
mais uma matéria!
Obrigado!

Ercules Mauricio

ercules.mauricio@t2mlab.com



Esta apresentação se destina exclusivamente à exibição interna e seu conteúdo não poderá ser divulgado à clientes ou público externo sem prévia autorização.



t2m
test to market