

Curso de Automação de Testes de Frontend Web com Python e Selenium

...

Aula 03

Paulo Oliveira

Agenda

- Revisão sobre locators
- Locators: CSS Selector e XPath
- Melhores práticas no uso de locators
- Asserts simples
- Interagindo com elementos

Locators - Revisão

Locator	Descrição	Sintaxe em Python
id	Identifica o WebElement usando o atributo ID.	<code>driver.find_element(By.ID, "IdValue")</code>
name	Identifica o WebElement usando o atributo Nome	<code>driver.find_element(By.NAME, "nameValue")</code>
className	Usa o atributo Classe para identificar o objeto	<code>driver.find_element(By.CLASS_NAME, "classValue")</code>
linkText	Usa o texto em hiperlinks para localizar o WebElement	<code>driver.find_element(By.LINK_TEXT, "textofLink")</code>
partialLinkText	Usa uma parte do texto em hiperlinks para localizar o WebElement	<code>driver.find_element(By.PARTIAL_LINK_TEXT, "PartialTextofLink")</code>
tagName	Usa o tagName para localizar o WebElement desejado	<code>driver.find_element(By.TAG_NAME, "htmlTag")</code>

Locators - CSS Selector e XPath

- CSS Selector
 - Definição: Uma sintaxe usada para identificar e selecionar elementos dentro de documentos HTML para aplicação de estilos.
 - Uso: Utilizado principalmente para identificar elementos usando regras de estilo em elementos específicos em páginas da web. Muito útil em automação de testes.
- XPath
 - Definição: Uma linguagem de consulta que permite a navegação e seleção de partes específicas dentro de documentos XML (e por extensão, HTML).
 - Uso: Usado para localizar nós em documentos XML de forma precisa, facilitando a manipulação, extração, e edição de dados. Muito útil em automação de testes e scraping de dados web.

Locators - CSS Selector e XPath

Locator	Descrição	Sintaxe em Python
xpath	Usa XPath para localizar o WebElement	<code>driver.find_element(By.XPATH, "xpathValue")</code>
cssSelector	CSS usado para criar regras de estilo na página web é aproveitado para localizar o WebElement desejado	<code>driver.find_element(By.CSS_SELECTOR, "cssValue")</code>

Locators

- Site usado nos exemplos:
 - <https://paulocoliveira.github.io/mypages/locators/index.html>
- `//*[@id="sidebar"]/ul/li[3]/a`

CSS Selector x Xpath

Scenarios	CSS Selector	Xpath
All elements	*	//*
Single element	nav	//nav
Single element with absolute path	html > body > div > nav	//html/body/div/nav
More than one element	header, footer	//header //footer
An element attribute	article[data-status="draft"]	//article[@data-status="draft"]
Part of an element attribute	article[data-status*="pub"]	//article[contains(@data-status,"pub")]
By class	.tags	//ul[@class="tags"]
By ID	#content	//section[@id="content"]
By text	-	//h2[text()="XPath in Test Automation"]
Elements hierarchy	#content h2	//section[@id="content"]/h2
Elements hierarchy (direct child)	#content > article > h2	//section[@id="content"]/article/h2
Specific childs	nav li:first-child nav li:nth-child(2) nav li:last-child	//nav//ul/li[1] //nav//ul/li[2] //nav//ul/li[last()]
Sibling	-	//nav//ul/li[2]/preceding-sibling::li //nav//ul/li[2]/following-sibling::li

Melhores Práticas no Uso de Locators

- Use um ID único para identificação do elemento
- Evite usar IDs que mudam de posição
- Mantenha seus localizadores curtos
- Evite elementos que dependem de valores gerados automaticamente
- Não use xpath ou Seletor CSS fornecidos pelas ferramentas de desenvolvedor
- Evite usar localizadores que mudam dinamicamente ao atualizar a página
- Evite usar xpath e css selector intensivamente no projeto

Asserts simples

- O que são?
 - Declarações que verificam se uma condição é verdadeira.
 - Utilizados para validar o resultado de um teste.
- Importância
 - Garantem que o código se comporta conforme esperado.
 - Identificam falhas e problemas no código testado.
- Exemplos
 - Verificar igualdade: `assert resultado == esperado`
 - Verificar verdadeiro/falso: `assert condição`
 - Verificar não nulo: `assert objeto is not None`
- Exemplos com Selenium
 - Verificar o título de uma página: `assert "Título esperado" in driver.title`
 - Verificar a presença de um elemento: `assert driver.find_element(By.ID, "elemento")`

Interagindo com elementos

- Text Boxes
 - Objetivo: Adicionar dados em texto
 - `send_keys()`
- Buttons
 - Objetivo: Submeter dados, navegar entre páginas, disparar funcionalidades
 - `click()`
- Checkboxes
 - Objetivo: Selecionar uma ou mais opção em uma lista pré-definida
 - `click()`
- Radio Buttons
 - Objetivo: Selecionar uma opção em uma lista pré-definida
 - `click()`
- Dropdown Menus
 - Objetivo: Selecionar uma opção em uma lista pré-definida
 - `select_by_visible_text()`
- Texto
 - Objetivo: Exibir um texto
 - `.text`

Locators

- Site usado nos exemplos:
 - <https://paulocoliveira.github.io/mypages/jobapplication.html>

Perguntas?

- Momento aberto para dúvidas



Próximos Passos

- O que esperar da próxima aula
 - Plugins Python para Testes (Pytest e Unittest) e Apresentação do Projeto Prático 1
- Prática de Casa
 - Refazer a automação do [Job Application Form](#) utilizando outros locators diferentes do que foi apresentado na aula
- Ganhe visibilidade todos os dias

Curso de Automação de Testes de Frontend Web com Python e Selenium

...

Aula 03

Paulo Oliveira