Testing II

Localizadores en Selenium

¿Qué son los localizadores en Selenium?

Los localizadores se definen como una **dirección que identifica de forma única un elemento web dentro de la página web**. Es un comando que le indica al IDE de Selenium qué elementos de la interfaz gráfica de usuario, como cuadros de texto, botones, casillas de verificación, etc., debe operar. Encontrar los elementos correctos de la GUI es un requisito previo para crear un script de automatización, pero identificar con precisión los elementos de la GUI es mucho más difícil de lo que parece. A veces, incluso se puede acabar trabajando con elementos de la interfaz gráfica de usuario incorrectos o sin ningún elemento. Así, el uso del buscador adecuado garantiza que las pruebas sean más rápidas, más fiables o de menor mantenimiento en comparación con las versiones.

Si tienen la suerte de trabajar con IDs y clases únicas, normalmente lo tienen todo listo. Pero habrá ocasiones en las que elegir el localizador adecuado se convierta en una pesadilla debido a la complejidad de encontrar los elementos de la página web.

Tipos de localizadores en Selenium

Hay una gama diversa de elementos web como caja de texto, id, botón de opción, etc. Se requiere un enfoque eficaz y preciso para identificar estos elementos. Así, se puede afirmar que con el aumento de la eficacia del localizador, aumentará la estabilidad del script de automatización.

Hay ocho estrategias diferentes de localización de elementos incorporadas en Selenium WebDriver:

Localizador	**Descripción**	**Sintaxis**
class name	Buscar elementos cuyo nombre de clase contenga el valor de búsqueda —no se permiten nombres de clase compuestos—.	driver.findElement(By.class Name (<element class>))
css selector	Buscar elementos que coincidan con un selector CSS.	driver.findElement(By.cssSe lector (<css selector>))
id	Encuentra los elementos cuyo atributo ID coincide con el valor de búsqueda.	driver.findElement(By.id (<element ID>))
name	Encuentra los elementos cuyo atributo NAME se corresponde con el valor de búsqueda.	driver.findElement(By.name (<element name>))
link text	Busca elementos de anclaje cuyo texto visible coincida con el valor de búsqueda.	driver.findElement(By.linkTe xt (<linktext>))

partial link text	Encuentra los elementos de anclaje cuyo texto visible contiene el valor de búsqueda. Si hay varios elementos coincidentes, sólo se seleccionará el primero.	driver.findElement(By.partialLinkText (<linktext>))
tag name	Busca elementos cuyo nombre de etiqueta coincida con el valor de búsqueda.	driver.findElement(By.tagName (<htmltagname>))
xpath	Busca elementos que coincidan con una expresión XPath.	driver.findElement(By.xpath (<xpath>))

Prácticas recomendadas para los localizadores de Selenium

Entender el concepto de localizadores en Selenium es una cosa, pero saber utilizarlos es otra muy distinta. Para poder **construir un localizador robusto** hay que entender qué es un localizador robusto.

A continuación se enumeran tres criterios que debe seguir cuando utilice localizadores en Selenium:

- Los localizadores robustos en Selenium son tan simples y pequeños como es posible: cuantos más elementos contenga un localizador, mayores serán las posibilidades de que se rompa debido a un cambio en la estructura de la página.
- Los localizadores de Selenium siguen funcionando después de cambiar las propiedades de un elemento de la interfaz de usuario: confiar en los atributos que se cambian con frecuencia, como las clases modificadoras (menu__item-red), nunca es una buena práctica.

- Los localizadores de Selenium, que son robustos por naturaleza, siguen funcionando después de que se cambien los elementos de la UI alrededor del elemento al que se dirige: siempre que se utilice un atributo no único, lo más probable es que los localizadores se rompan porque alguien haya añadido un elemento con el mismo atributo por encima.