Selenium 4 Beginners

Rex Jones II

Find WebElements Using Selenium Locators

Table of Contents

Introduction 2

Demo 2

Selenium Locator Ranks.. 3

Introduction

Hello and Welcome To Selenium 4 Beginners. We are going to discuss How To Use Selenium Locators To Find WebElements. The Transcript, Presentation, and Code will be available on github at RexJonesII/Selenium4Beginners and https://tinyurl.com/SeleniumLocatorsForWebElements.

Before checking out How To Use A Selenium Locator To Find A WebElement, let's look at the Selenium Locators then at the end we are going to look at the rankings for the Selenium Locators.

In alphabetical order, the 8 Selenium Locators are className, cssSelector, id, linkText, name, partialLinkText, tagName, and xpath. Now, let's look at the Selenium Locators that help us find WebElements.

Demo

I believe finding a WebElement is the most important aspect of automation. Why? because we are unable to perform any action on a WebElement until we first find the WebElement.

I have already created an instance of WebDriver, a setup method to load OrangeHRM, and our Test Method called demoSeleniumLocators. Recall from the previous video, I mentioned id, name, and class are the key attributes for finding WebElements. Therefore, I will start with id which is a unique identifier for an element. We write driver.findElement By then the dot operator. Here's a list of all 8 Selenium Locators, select id, (pause) and the parameter is the value of id attribute. Let's go to the AUT and find the value of id by inspecting username. The value is txtUsername. Copy the value and paste the value. WebElement username = We are not going to perform actions on the WebElement but only find the WebElement. As a result, I will demonstrate finding the WebElement by highlighting each WebElement. Highlighter.highlightElement pass in the driver and pass in username as the WebElement. This is a static highlighter method so I can call it directly from the utility package. The background will be red with a black border. Highlighter methods are good for demos because it can slow down execution. This execution will slow down for 1 second due to Thread.sleep 1000. Let's run. Did you see Username highlight red? That shows we successfully found Username. We are going to do that for each WebElement.

driver.findElement By dot name and the parameter is the value of name attribute Next, is the Password field. The name attribute has a value of txtPassword. Copy and Paste the value. WebElement password = Highlighter.highlightElement pass in the driver and password. Let's run again then see username and password highlight red.

driver.findElement By dot className and the description shows Find elements based on the value of the class attribute. Inspect button. The class attribute has a value of button. Copy and Paste the value. WebElement buttonLogin = Highlighter.highlightElement pass in the driver and buttonLogin. Let's Run. We saw all 3 WebElements show a red background.

The next 2 Selenium Locators help find links on a web page: linkText and partialLinkText driver.findElement By dot linkText and parameter is the exact text to match against driver.findElement By dot partialLinkText finds part of the text.

Let’s use Forgot your password for linkText and OrangeHRM, Inc for partialLinkText. Inspect both WebElements. Here’s the text for Forgot your password and OrangeHRM. WebElement linkForgotPassword = Highlighter.highlightElement pass in driver and linkForgotPassword

The value for Partial Link Text will be OrangeHRM. WebElement linkOrangeHRM = Highlighter.highlightElement pass in driver and linkOrangeHRM. Let’s scroll down the page before Partial Link Text.

The next Selenium Locator is Tag Name. We are going to find all of the Tag Names for an image. I’m going to move Chrome DevTools. The locator CTRL + F can help us see how many tags have img. 2 forward slashes img. There are 5 tags with img. Let’s look at the 5 images. The first image is at the top OrangeHRM and the next 4 images are at the bottom: LinkedIn, Facebook, Twitter, and YouTube.

Go back to Eclipse and this time, we type driver.findElements with an s because we are looking for more than 1 WebElement. The description shows find all elements within the current page. Notice the syntax for assigning all WebElements is List angle brackets with WebElement in between the brackets. We continue with

By dot tagName double quotes img. List then the angle brackets WebElement images = This time, we loop through all of the images by writing for (WebElement image : images) images is a list of all images. On the other hand, image is a temporary variable. Therefore, we write, Highlighter.highlightElement pass in driver and image. Now, let’s run.

Selenium Locator Ranks

Let’s look the ranking for Selenium Locators. id is Number 1 because it’s a unique identifier for an element. Number 2 is name. Most of the times, name is unique for the web page. I have seen cases where id and name are unique for an element but not unique for a page. If an id or name attribute is not available then third is class name. A class name does not have to be unique. There can be 1 or more class names for an element. Some elements may not have a class name at all. Link Text and Partial Link Text are good but can lead to flaky values because text on a page can change. Here’s an example of the language changing from English to Spanish. I showed you Tag Names but they are not effective if you’re looking 1 element because it returns all elements. Therefore, if text can change on a page and it’s also possible for id, name, or class to not be available. What should we use? We should use anyone of the most powerful Selenium Locators: XPath or CSS Selectors.

Next, we will look at the XPath Selenium Locator.