Rex Jones II

Mouse Double Click

Table of Contents

Introduction

Hello and Welcome. My name is Rex Jones II. If you have not subscribed to my YouTube channel, click the Subscribe button then bell icon for each video notification. Also, connect with me on LinkedIn. In this video, we are going to automate double clicking a box using Selenium's Actions class.

jQuery is the AUT. I'm going to scroll to the bottom of this AUT and we see this blue box. The Test Script will get the color of the box, double click the box, then get the new color of the box. Inspect and we see View Frame Source shows us the box is located in an iframe. First, let's find the iframe by writting type 2 slashes, iframe. The application only has 1 frame. That makes it easy for us.

Demo

Go to Eclipse and the first thing we do is scroll to the bottom of the page using JavaScript Executor. Now, switch to the frame. driver.switchTo().frame It's best to switch to a frame using WebElement, name, or id but we only have 1 frame. So, let's use the index. (0); You can watch Videos 29 – 32 to learn more about switching to a frame.

After switching to the frame, we find the box but first let's import JavaScriptExecutor. Go back and inspect the box.

Within the iframe element, we can find the box using a direct relationship between the parent and child. body is the parent and div is the child. Therefore, we write body angle bracket div. There are 4 results but 3 of the results are outside the iframe. Let's search, search, search. 4 of 4 contains our element.

Let me show you how we get the color since we are inspecting the box. Select the Computed tab. Do you see how the background-color is yellow? We will use background-color as a parameter for the getCssValue Selenium method. The method will return the computed value which is 255, 255, 0.

Watch how the color changes after reloading the page and inspecting the box. It changed back to blue and we see rgb updated to 0, 0, 255.

Back to Eclipse. Find the box WebElement box = driver.findElement(By.*cssSelector*("body > div")); Now print the color.
Sysout System.*out*.println("Color Before: " + box.getCssValue We see the description shows Color values should be returned as rgba strings. RGBA stands for red green blue alpha. It's a 4-channel color model. The parameter - propertyName is background-color surrounded by double quotes.

Now, we double click by instantiating the Actions class act = new Actions (driver);
act.doubleClick and the WebElement target is box then we perform the action. The last step is to print color after double clicking the box. Copy and Paste then change Before to After. Let's Run. We see the colors are different before and after double clicking the box. That's it and Thank You for watching How To Double Clicking A WebElement Using Selenium Actions Class.