Double-Click and Right-Click

In this lesson, we will discuss how to handle double-click and right-click using the Selenium WebDriver.

The below example code can be downloaded and run locally; it is included as a part of the Sample UI test framework, please refer.

Selenium provides a predefined `org.openqa.selenium.interactions.Actions` class used to perform multiple keyboard and mouse operations such as Right-click, Double-click, etc.

Double-click

```java
// Create a WebDriver object
WebDriver driver = DriverManager.getWebDriver();

driver.get("http://codetoautomate.com/educative-selenium-demo/");

// scroll to the end of the page
JavascriptExecutor jsDriver = (JavascriptExecutor) driver;
jsDriver.executeScript("document.getElementById('dbl-click').scrollIntoVie
wIfNeeded()");

// Find the element and perform an operation
WebElement source = driver.findElement(By.id("dbl-click"));

Actions actions = new Actions(driver);
actions.doubleClick(source).build().perform();

// Wait for the element
WebDriverWait wait = new WebDriverWait(driver, 30);
Alert alert = wait.until(ExpectedConditions.alertIsPresent());
```

```
Assert.assertNotNull(alert, "alert on double-click did not appear");


alert.accept();
```

Right-click

```
// create a WebDriver object
WebDriver driver = DriverManager.getWebDriver();

// Open the web URL
driver.get("http://codetoautomate.com/educative-selenium-demo/");

WebElement source = driver.findElement(By.id("prompt"));
Actions actions = new Actions(driver);

actions.contextClick(source).build().perform();
```

That's all for double-clicking and right-clicking. In the next lesson, you'll learn how to capture a screenshot of a webpage using Selenium.