Selenium WebDriver- Drag and Drop

In this section, you will learn how to perform complex operation like Drag and Drop in Selenium WebDriver.

Before proceeding with this section, let us first understand some of the concepts regarding Drag and Drop operation.

Actions in Selenium WebDriver

For performing complex user interaction like drag and drop, we have an Actions class in Selenium WebDriver. Using the Actions class, we first build a sequence of composite events and then perform it using Action (an interface which represents a single user-interaction). The different methods of Actions class we will be using here are-

* **clickAndHold(WebElement element)** - Clicks a web element at the middle(without releasing).
* **moveToElement(WebElement element)** - Moves the mouse pointer to the middle of the web element without clicking.
* **release(WebElement element)** - Releases the left click (which is in pressed state).
* **build()** - Generates a composite action

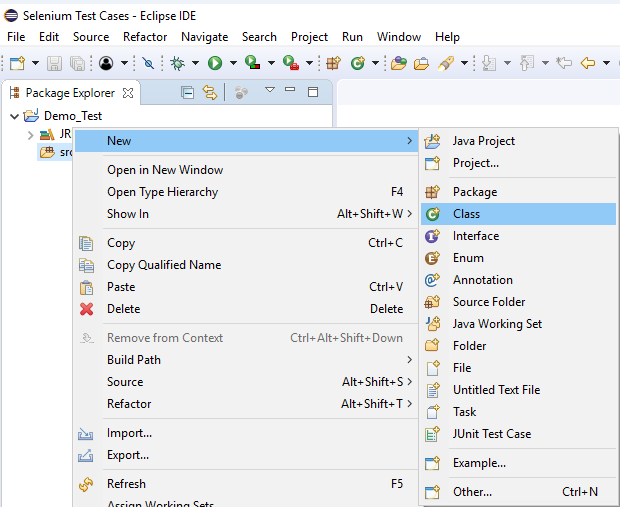
Let us consider a test case in which we will automate the following scenarios:

* Invoke Firefox Browser
* Open URL: <https://www.testandquiz.com/selenium/testing.html>
* Drag and Drop the JavaTpoint icon on the textbox

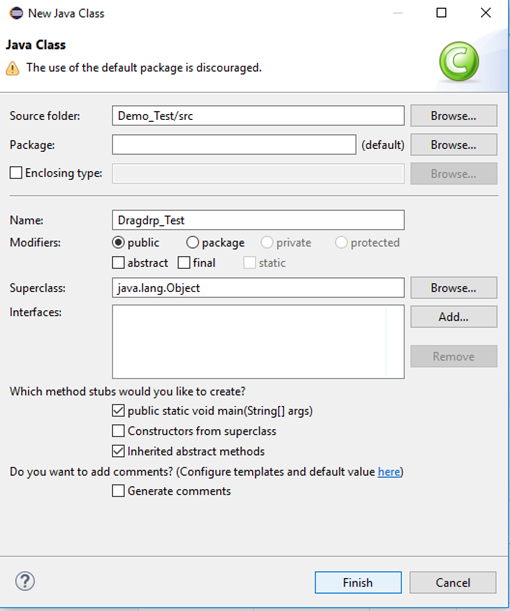
We will create our test case step by step in order to give you a complete understanding of how to handle drag and drop in WebDriver.

**Step1**. Launch Eclipse IDE and open the existing test suite "Demo\_Test" which we have created in earlier sessions of this tutorial.

**Step2**. Right click on the "src" folder and create a new Class File from **New >Class**.



Give your Class name as "Dragdrp\_Test" and click on "Finish" button.



**Step3**. Let's get to the coding ground.

* To invoke Firefox browser, we need to download Gecko driver and set the system property for Gecko driver. We have already discussed this in earlier sessions of this tutorial. You can refer ["Running test on Firefox Browser"](https://www.javatpoint.com/selenium-webdriver-running-test-on-firefox-browser-gecko-driver) to learn how to download and set System property for Firefox driver.

Here is the sample code to set system property for Gecko driver:

1. // System Property for Gecko Driver
2. ystem.setProperty("webdriver.gecko.driver","D:\\GeckoDriver\\geckodriver.exe" );

* After that we have to initialize Gecko Driver using Desired Capabilities Class.

Here is the sample code to initialize gecko driver using DesiredCapabilities class.

1. // Initialize Gecko Driver using Desired Capabilities Class
2. DesiredCapabilities capabilities = DesiredCapabilities.firefox();
3. capabilities.setCapability("marionette",**true**);
4. WebDriver driver= **new** FirefoxDriver(capabilities);

Combining both of the above code blocks, we will get the code snippet to launch Firefox browser.

1. // System Property for Gecko Driver
2. System.setProperty("webdriver.gecko.driver","D:\\GeckoDriver\\geckodriver.exe" );
4. // Initialize Gecko Driver using Desired Capabilities Class
5. DesiredCapabilities capabilities = DesiredCapabilities.firefox();
6. capabilities.setCapability("marionette",**true**);
7. WebDriver driver= **new** FirefoxDriver(capabilities);

After that we need to write the code which will automate our second test scenario (navigate to the desired URL)

Here is the sample code to navigate to the desired URL:

1. // Launch Website
2. driver.navigate().to("https://www.testandquiz.com/selenium/testing.html");

The complete code till now will look something like this:

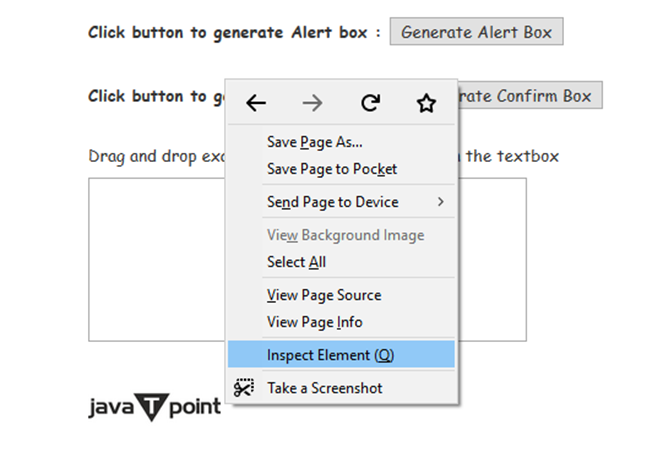
1. **import** org.openqa.selenium.WebDriver;
2. **import** org.openqa.selenium.firefox.FirefoxDriver;
3. **import** org.openqa.selenium.remote.DesiredCapabilities;
5. **public** **class** Dragdrp\_Test {
7. **public** **static** **void** main(String[] args) {
9. // System Property for Gecko Driver
10. System.setProperty("webdriver.gecko.driver","D:\\GeckoDriver\\geckodriver.exe" );
12. // Initialize Gecko Driver using Desired Capabilities Class
13. DesiredCapabilities capabilities = DesiredCapabilities.firefox();
14. capabilities.setCapability("marionette",**true**);
15. WebDriver driver= **new** FirefoxDriver(capabilities);

18. // Launch Website
19. driver.navigate().to("https://www.testandquiz.com/selenium/testing.html");
21. }
23. }

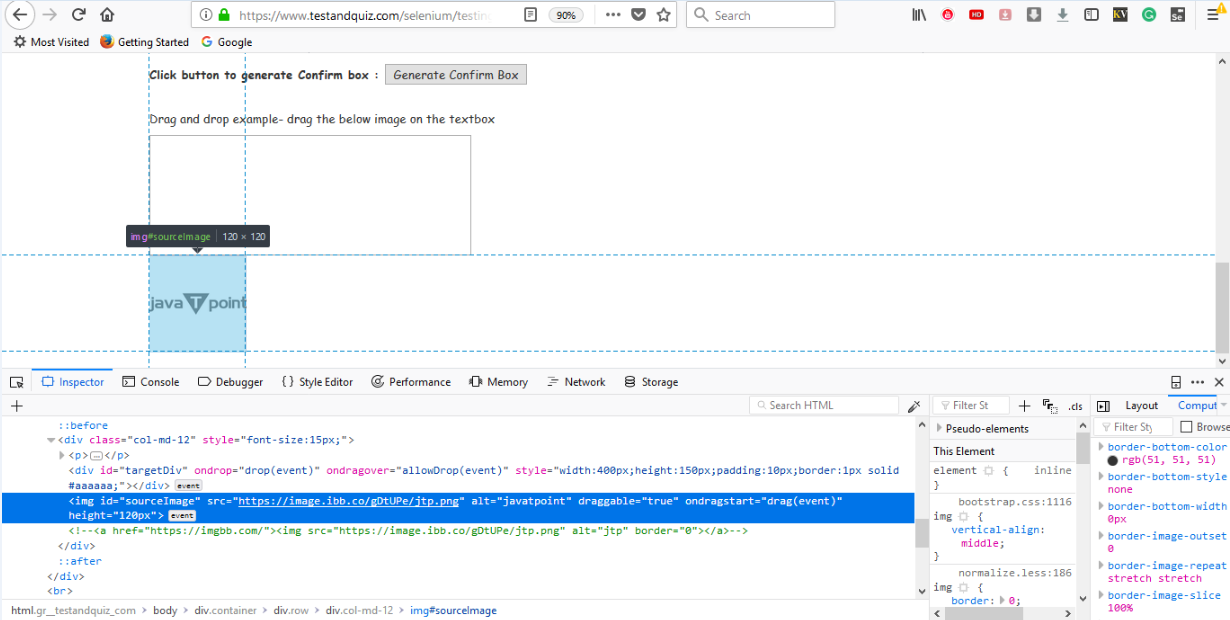
**Step4**. Now we will try to locate the JavaTpoint icon and text box in order to perform drag and drop operation. As we know that locating an element involves inspection of its HTML codes.

Follow the steps given below to locate the drop-down menu on the sample web page.

* Open URL: <https://www.testandquiz.com/selenium/testing.html>
* Right click on JavaTpoint icon and select Inspect Element.



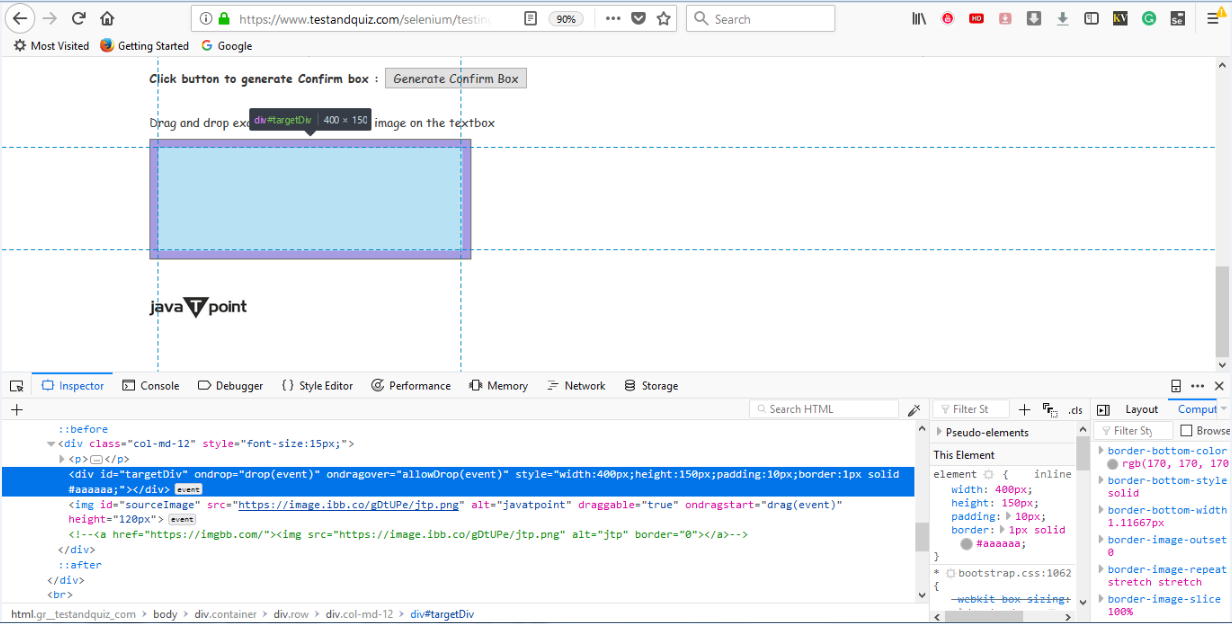
* It will launch a window containing all the specific codes involved in the development of the JavaTpoint logo.



* Take a note of its id attribute i.e. sourceImage



Similarly, we will inspect the text box where we have to place the logo.



* Take a note of its id attribute i.e. targetDiv



**Step5.** To automate our third and fourth test scenario, we need to write the code which will perform the drag and drop operation on the JavaTpoint logo.

Given below is the code snippet to perform drag and drop operation.

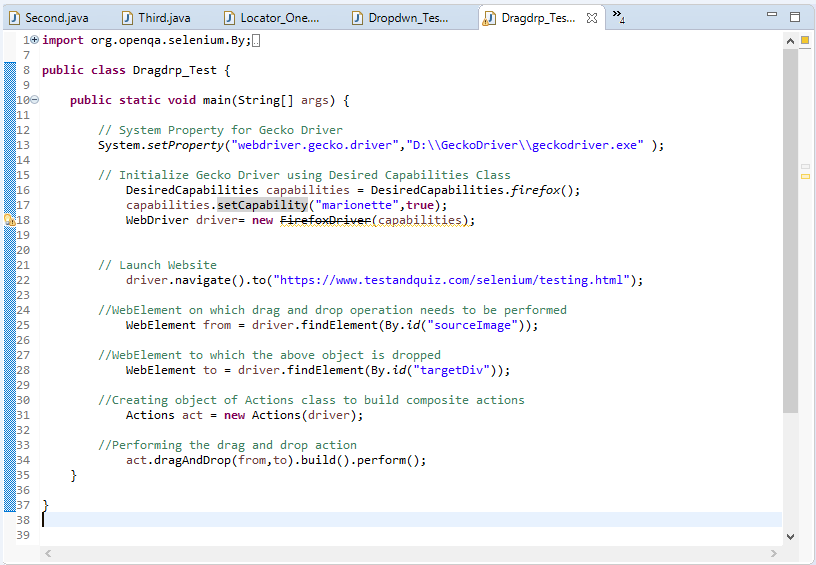
1. //WebElement on which drag and drop operation needs to be performed
2. WebElement from = driver.findElement(By.id("sourceImage"));
4. //WebElement to which the above object is dropped
5. WebElement to = driver.findElement(By.id("targetDiv");
7. //Creating object of Actions class to build composite actions
8. Actions act = **new** Actions(driver);
10. //Performing the drag and drop action
11. act.dragAndDrop(from,to).build().perform();

Thus, our final test script will look something like this:

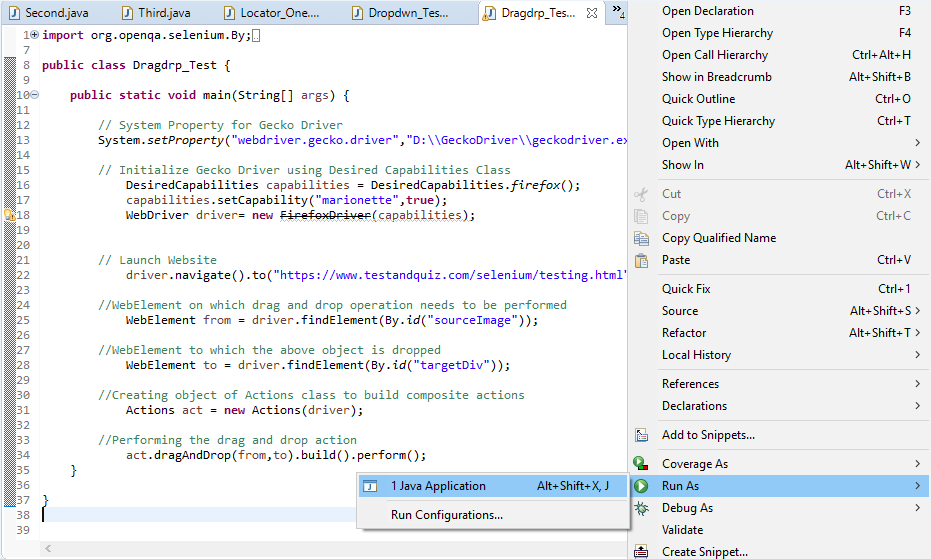
1. **import** org.openqa.selenium.By;
2. **import** org.openqa.selenium.WebDriver;
3. **import** org.openqa.selenium.WebElement;
4. **import** org.openqa.selenium.firefox.FirefoxDriver;
5. **import** org.openqa.selenium.interactions.Actions;
6. **import** org.openqa.selenium.remote.DesiredCapabilities;
8. **public** **class** Dragdrp\_Test {
10. **public** **static** **void** main(String[] args) {
12. // System Property for Gecko Driver
13. System.setProperty("webdriver.gecko.driver","D:\\GeckoDriver\\geckodriver.exe" );
15. // Initialize Gecko Driver using Desired Capabilities Class
16. DesiredCapabilities capabilities = DesiredCapabilities.firefox();
17. capabilities.setCapability("marionette",**true**);
18. WebDriver driver= **new** FirefoxDriver(capabilities);

21. // Launch Website
22. driver.navigate().to("https://www.testandquiz.com/selenium/testing.html");
24. //WebElement on which drag and drop operation needs to be performed
25. WebElement from = driver.findElement(By.id("sourceImage"));
27. //WebElement to which the above object is dropped
28. WebElement to = driver.findElement(By.id("targetDiv"));
30. //Creating object of Actions class to build composite actions
31. Actions act = **new** Actions(driver);
33. //Performing the drag and drop action
34. act.dragAndDrop(from,to).build().perform();
35. }
37. }

The following screenshot shows the Eclipse window for our test script.



**Step6**. Right click on the Eclipse code and select **Run As > Java Application**.



Upon execution, the above test script will launch the Firefox browser and automate all the test scenarios.

You can see that the JavaTpoint logo will automatically get dropped into the text box.

