Selenium WebDriver- Handling drop-downs

In this section, you will learn how to handle drop-downs in Selenium WebDriver.

Before proceeding with this section, let us first understand some of the basics of handling drop-downs in Selenium WebDriver.

Select in Selenium WebDriver

The 'Select' class in Selenium WebDriver is used for selecting and deselecting option in a dropdown. The objects of Select type can be initialized by passing the dropdown webElement as parameter to its constructor.

1. WebElement testDropDown = driver.findElement(By.id("testingDropdown"));
2. Select dropdown = **new** Select(testDropDown);

**How to select an option from drop-down menu?**

WebDriver provides three ways to select an option from the drop-down menu.

**1. selectByIndex** - It is used to select an option based on its index, beginning with 0.

1. dropdown.selectByIndex(5);

**2. selectByValue** - It is used to select an option based on its 'value' attribute.

1. dropdown.selectByValue("Database");

**3. selectByVisibleText** - It is used to select an option based on the text over the option.

1. dropdown.selectByVisibleText("Database Testing");

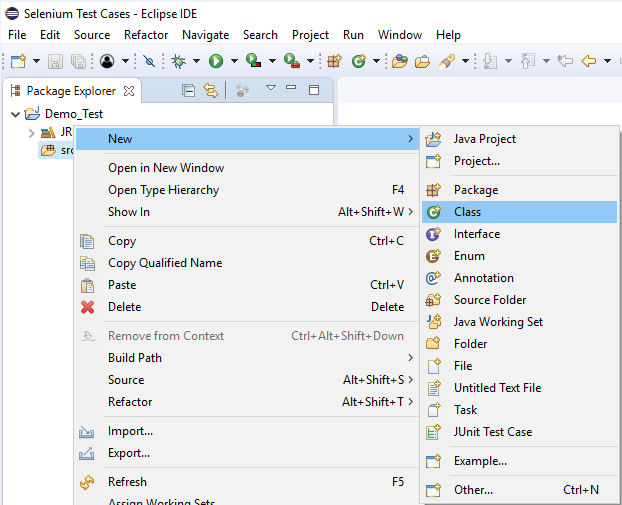
Let us consider a test case in which we will automate the following scenarios:

* Invoke Google Chrome Browser
* Open URL: <https://www.testandquiz.com/selenium/testing.html>
* Select the option "Database Testing" from the drop-down menu
* Close the browser

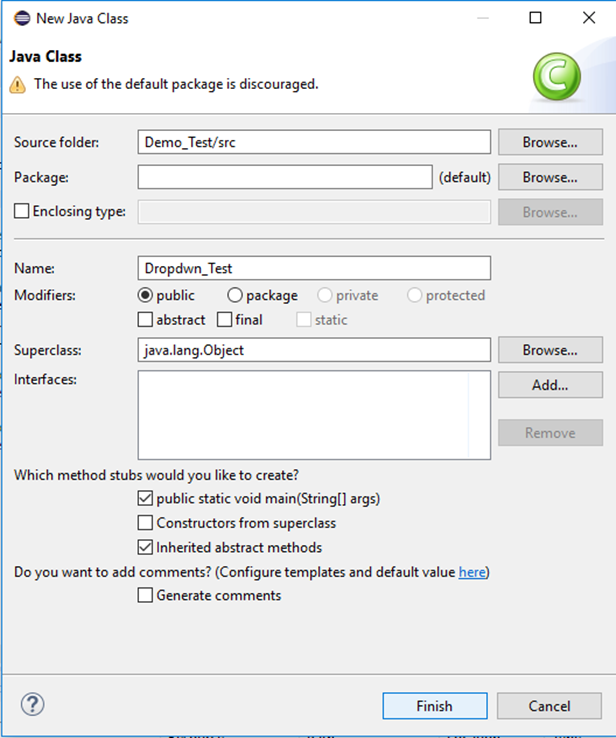
We will create our test case step by step in order to give you a complete understanding of how to handle drop-downs in WebDriver.

**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 "Dropdwn\_Test" and click on "Finish" button.



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

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

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

1. // System Property for Chrome Driver
2. System.setProperty("webdriver.chrome.driver","D:\\ChromeDriver\\chromedriver.exe");

* After that we have to initialize Chrome driver using ChromeDriver Class.

Here is the sample code to initialize Chrome driver using ChromeDriver class.

1. // Instantiate a ChromeDriver class.
2. r driver=**new** ChromeDriver();

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

1. // System Property for Chrome Driver
2. m.setProperty("webdriver.chrome.driver","D:\\ChromeDriver\\chromedriver.exe");
3. // Instantiate a ChromeDriver class.
4. river driver=**new** ChromeDriver();

* 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.chrome.ChromeDriver;

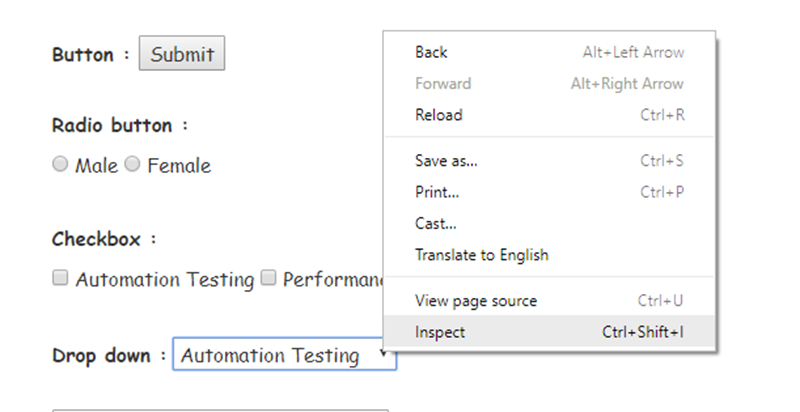

6. **public** **class** Partial\_Link {
8. **public** **static** **void** main(String[] args) {
10. // System Property for Chrome Driver
11. System.setProperty("webdriver.chrome.driver","D:\\ChromeDriver\\chromedriver.exe");
13. // Instantiate a ChromeDriver class.
14. WebDriver driver=**new** ChromeDriver();
16. // Launch Website
17. driver.navigate().to("https://www.testandquiz.com/selenium/testing.html");


21. }
23. }

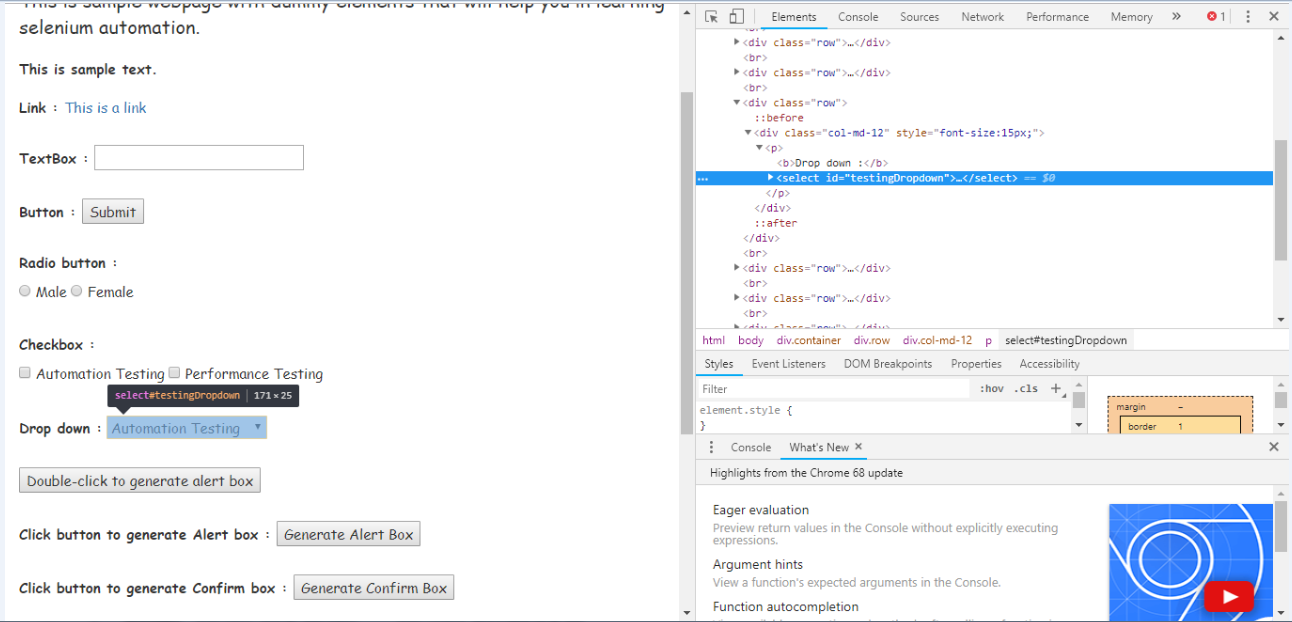
**Step4**.Now we will try to locate the drop-down menu by inspecting its HTML codes.

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

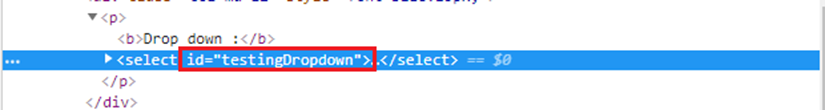
* Open URL: <https://www.testandquiz.com/selenium/testing.html>
* Right click on the drop-down menu on the sample web page and select Inspect Element



* It will launch a window containing all the specific codes involved in the development of the drop-down menu.



* Take a note of its id attribute.



**Step5**. To automate our third test scenario, we need to write the code which will select the option "Database Testing" from the drop-down menu.

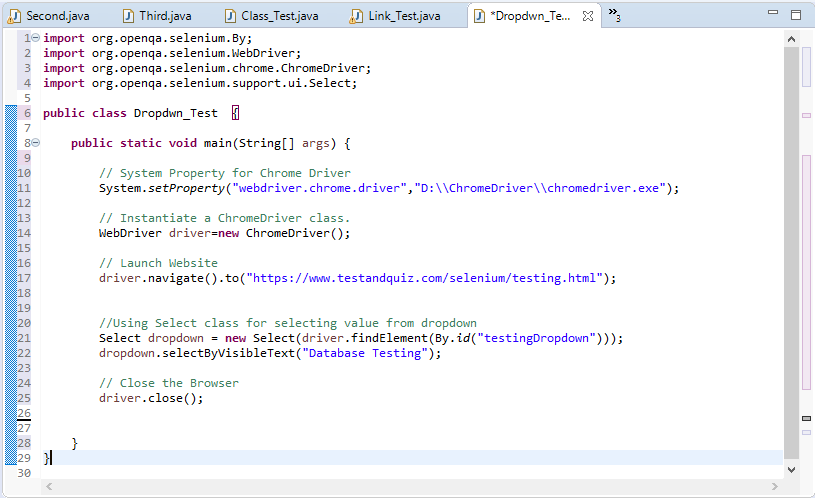
Here is the sample code to so that:

1. //Using Select class for selecting value from dropdown
2. Select dropdown = **new** Select(driver.findElement(By.id("testingDropdown")));
3. dropdown.selectByVisibleText("Database Testing");
5. Thus, our **final** test script will look something like **this**:
6. **import** org.openqa.selenium.By;
7. **import** org.openqa.selenium.WebDriver;
8. **import** org.openqa.selenium.chrome.ChromeDriver;
9. **import** org.openqa.selenium.support.ui.Select;
11. **public** **class** Dropdwn\_Test  {
13. **public** **static** **void** main(String[] args) {
15. // System Property for Chrome Driver
16. System.setProperty("webdriver.chrome.driver","D:\\ChromeDriver\\chromedriver.exe");
18. // Instantiate a ChromeDriver class.
19. WebDriver driver=**new** ChromeDriver();
21. // Launch Website
22. driver.navigate().to("https://www.testandquiz.com/selenium/testing.html");

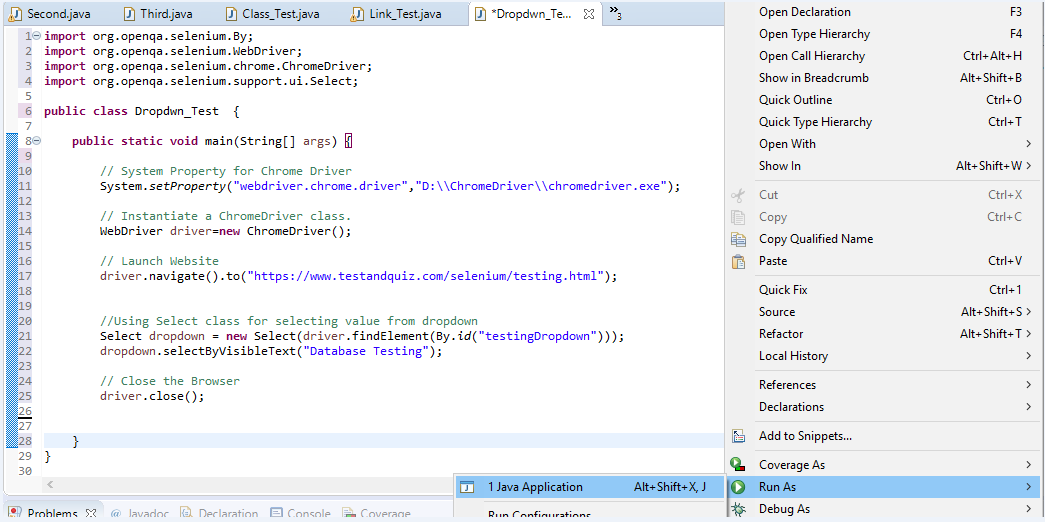
25. //Using Select class for selecting value from dropdown
27. Select dropdown = **new** Select(driver.findElement(By.id("testingDropdown")));
28. dropdown.selectByVisibleText("Database Testing");
30. // Close the Browser
31. driver.close();

34. }
35. }

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 Chrome browser and automate all the test scenarios.