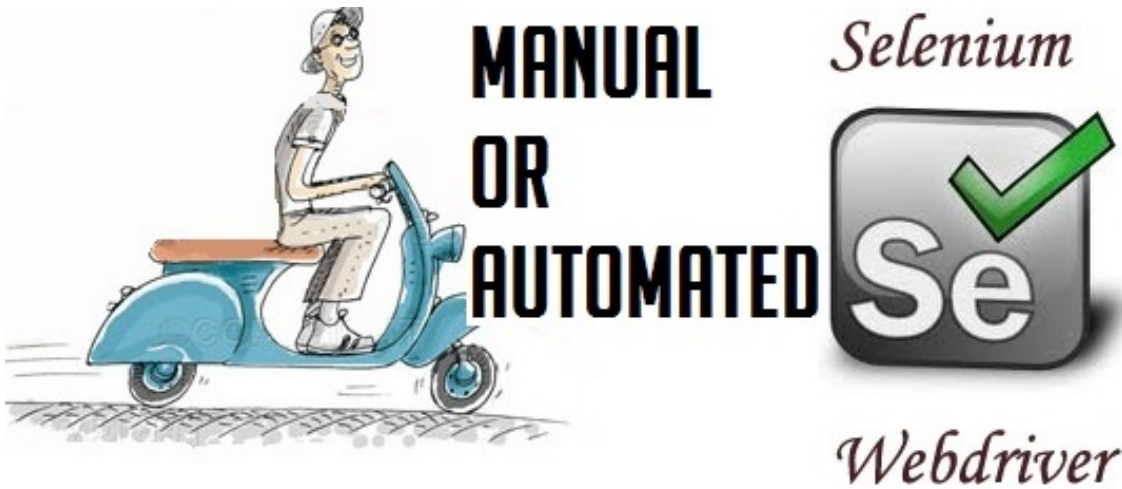


15 Most Commonly Used Selenium WebDriver Command's in Java



19

SHARE THIS

Share

Tweet

Selenium 2.0 also known as WebDriver is the best automation tool for web and offers very user friendly learning and impressive work. WebDriver has covered all the limitation of Selenium RC and work on a standalone server. Here is the basic and most commonly used functions of WebDriver with Java.

Popup Windows and Frames

Popup window handling is one of the must handle things in Automation and Webdriver does is really well. Using the window handle command.

```
1 String winHandleBefore = driver.getWindowHandle();
2 for(String winHandle : driver.getWindowHandles()){
3     driver.switchTo().window(winHandle);
4 }
```

Once action is performed, switch back to parent window.

```
1 driver.switchTo().window(driver.getWindowHandle());
```

List Box

Select an item in Listbox using Selenium Webdriver is quite easy, there two way's to do it.

```
1 WebElement select = driver.findElement(By.id("selection"));
2 List<WebElement> options = select.findElements(By.tagName("option"));
3 for (WebElement option : options) {
4     if("Germany".equals(option.getText()))
5         option.click();
6 }
```

Or

```
1 WebElement roleDropdown = driver.findElement(By.id(""));
2 roleDropdown.click();
```

Then

```
1 WebElement roleOptionOne = driver.findElement(By.id(FIRST_OPTION));
2 roleOptionOne.click();
```

Right Click

```
1 WebElement webelement = driver.findElement(By.xpath("*Xpath Locator*"));
2
3 Actions actions = new Actions(driver);
4 Action action = actions.contextClick(webelement).build();
5 action.perform();
6 <h2>Double Click</h2>
7 Actions action = new Actions(driver);
8 action.doubleClick(myElemment);
9 action.perform();
```

Excel Reader

Read or write data from excel using the jxl or POI, I will prefer POI. Create a property reader file then read the data

```
1 Workbook workbook = WorkbookFactory.create(new FileInputStream(file));
2 Sheet sheet = workbook.getSheetAt(0);
```

Database Connection

To connect to Database using WebDriver with Java, we use DBC("Java Database Connectivity) API.

```
1 DriverManager.getConnection(URL, "username", "password" )
```

Time Outs – Wait

Implicit Waits:

```
1 driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

Or

```
1 Thread.sleep(1000);
```

Explicit Waits

```
1 WebDriverWait wait = new WebDriverWait(driver, 10);  
2 WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id("someid")));  
3 <h2>Alert Box</h2>  
4 driver.findElement(By.id("updateButton")).click();  
5 Alert alert = driver.switchTo().alert();
```

`alert.accept(); or alert.dismiss();`

Navigation

```
1 driver.navigate().to(www.google.com);  
2 driver.get("http://www.google.com");  
3 driver.navigate().forward();  
4 driver.navigate().back();  
5 driver.navigate().refresh();  
6 driver.manage().deleteAllCookies();  
7 driver.close();
```

Table Columns

```
1 WebElement Table = driver.findElement(By.id(""));
2 List<WebElement> Row = Table.findElements(By.tagName("tr"));
```

Drag And Drop

```
1 WebElement element = driver.findElement(By.name("source"));
2 WebElement target = driver.findElement(By.name("target"));
3
4 (new Actions(driver)).dragAndDrop(element, target).perform();
```

Mouse Over

```
1 Actions actions = new Actions(driver);
2 WebElement mouseover = driver.findElement(By.id(""));
3 actions.moveToElement(mouseover);
4 actions.click().perform();
```

IsElement/TextPresent

There are various way to check whether an element is present or not before execution of a piece of code.

Option 1

```
1 private boolean isElementPresent(WebDriver driver, By by){
2 try{
3 driver.findElement(by);
4 return true;
5 }catch(NoSuchElementException e){
6 return false;
7 }
8 }
```

Option 2

```
1 if (driver.findElements(locator).size() > 0) {
2 return true;
3 } else {
4 return false;
5 }
6 }
```

Option 3

Sweet and simple one line of code

```
1 WebElement elementpresent = driver.PageSource.Contains("Sign out")
```

Web Elements Finding

These are generic ways to do certain common task i.e.

Id: `WebElement element = driver.findElement(By.id(""));`

Name: `WebElement element = driver.findElement(By.name(""));`

Tag Name: `WebElement frame = driver.findElement(By.tagName("iframe"));`

Xpath: `WebElement element = driver.findElement(By.xpath.name(""));`

CSS: `WebElement element = driver.findElement(By.CSS.name(""));`

LinkText: `WebElement element = driver.findElement(By.LinkText(""));`

I hope these commands will let you work with Selenium WebDriver like a star. Please leave us a comment and share your own experiences, commands with us.