

<https://tutorialsninja.com/demo/>
<https://omayo.blogspot.com/>

Save Copy to Evernote

Last updated: Jan 30, 2022

Selenium WebDriver Commands - Getting Started List (Selenium 4 - Session 11)

Selenium WebDriver is an API, which contains a set of predefined library of commands / methods.

- Google Search "Selenium Java WebDriver API" and find the predefined methods of WebDriver interface

Lets start using the Predefined methods of Selenium WebDriver.

- **get()**
 - Used to open the specified URL's web page
 - **http://** needs to be provided, otherwise the specified URL wont get opened
- **manage().window().maximize()**
 - Used to maximize the current web page
- **manage().window().minimize()**
 - Used to minimize the current web page and execute the script
- **findElement()**
 - **By** class and its predefined methods
 - **id()** - By.id("confirm")
 - **name()** - By.name("q")
 - **className()** - By.className("classone")
 - **linkText()** - By.linkText("compendiumdev")
 - **partialLinkText()** - By.partialLinkText("compendium")
 - **cssSelector()** - By.cssSelector("#confirm")
 - **xpath()** - By.xpath("//input[@id='confirm']")
 - **WebElement**
 - Predefined interface in Selenium
 - The purpose of using WebElement
 - Store the UI element located by the findElement() to reference variable of **WebElement** interface
- **click()**
 - Used to perform click operation on different UI elements like Button, link, checkbox option and radio option etc
- **sendKeys()**
 - Used to enter text into the text fields like text box, text area, password fields etc.
- **clear()**
 - Used to clear the text available in the text box or text area fields
- **getText()** - used to retrieve the element text(the text between two html tag)
 - Used to retrieve the elements text (i.e. The text between the starting and ending tags

[Terms of Service](#)[Privacy Policy](#)[Report Spam](#)[Save Copy to Evernote](#)

◦ Used to retrieve the title of the current web page

- **getCurrentUrl()**
 - Used to retrieve the URL of the current web page
- **close()**
 - Used to close the current Browser window
- **quit()**
 - Used to close all the Browser windows (i.e. All the browser windows, including child windows will be closed)
- **getAttribute()**
 - Used to retrieve the value stored in the specified attribute 'value' of the html element
- **isDisplayed()**
 - Used to find out whether the element is displayed on the page (i.e. available on the page) before performing operations on it
- **isEnabled()**
 - Used to find out whether the element is enabled or disabled before performing operation on it
- **isSelected()**
 - Used to find out whether the radio options and checkbox options are selected or not.
- **navigate()**
 - Used to perform operations like navigate back to previous page, navigate forward again or refreshing the current
- **getPageSource()**
 - Used to retrieve all the source code of the current page and return in the form of String
- **submit()**
 - Used to submit a form - Example: Search text field and Search Button on Omayo
- **getTagName()**
 - Used to get the html tag of the provided element - Example: find tag name of Search text field
- **getCSSValue()**
 - `driver.findElement(By.id("home")).getCssValue("line-height");`
- **getSize()**
 - Used to get the height and width of the given element
 - `Dimension d =driver.findElement(By.id("but2")).getSize();`
 - `d.height` and `d.width`
- **getLocation()**
 - Used to get the x and y coordinate position of the given element
 - `Pointp=driver.findElement(By.id("but2")).getLocation();`
 - `p.x` and `p.y`
- **getRect()**
 - Used to get the x and y coordinate position of the given element, along with height and width of the element
 - `Rectangle rectangle = driver.findElement(By.id("but2")).getRect()`
 - `rectangle.getX()`

- `rectangle.getY()`
- `rectangle.getWidth()`
- `rectangle.getHeight()`
- **fullScreen()**
 - Used to display the web page in full screen mode
 - `driver.manage().window().fullScreen();`
- **setSize()**
 - Used to resize the browser window in the specified dimensions
- **getClass()**
 - Used to retrieve the Class name of the provided object
 - `driver.getClass().getSimpleName()`
- **pageLoadTimeout()**
 - Used to change the default time for loading web page
 - `driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(3));`

The below are the list of Selenium WebDriver commands, which I will be explaining later as per the need basis:

- **findElements()** - this will return list of web elements
 - **By.tagName()**
 - **getScreenshotAs()**
 - **getFullPageScreenshotAs()**
 - **getWindowHandle()**
 - **getWindowHandles()**
 - **window()**
 - **newWindow()**
 - **switchTo()**
 - **selectByVisibleText()**
 - **selectByIndex()**
 - **selectByValue()**
 - **isMultiple()**
 - **getOptions()**
 - **getFirstSelectedOption()**
 - **getAllSelectedOptions()**
 - **deselectByVisibleText()**
 - **deselectByIndex()**
 - **deselectByValue()**
 - **deselectAll()**
 - **frame()**
 - **parentFrame()**
 - **defaultContent()**
 - **implicitlyWait()**
 - **until()**
 - **visibilityOfElementLocated()**
 - **elementToBeClickable()**
 - **invisibilityOfElementLocated()**
- ```
// Launch Browser - (Chrome)
// ChromeDriver driver = new ChromeDriver();
// WebDriver driver = new ChromeDriver();

// maximize size of page
driver.manage().window().maximize();

// Open URL https://demo.opencart.com/
driver.get("https://tutorialsninja.com/demo/");

// Validate title "Your Store"
String act_title = driver.getTitle();

if(act_title.equals("Your Store"))
 System.out.println("FirstDemo.main() | Test Passed");
else
 System.out.println("FirstDemo.main() | Test Failed....");

Thread.sleep(2000);

// return web element
WebElement element = driver.findElement(By.id("search"));

element.sendKeys("HP");
driver.findElement(By.className("input-group-btn")).click();
Thread.sleep(2000);

// Close Browser
// driver.close();
driver.quit();
```

- **invisibilityOfElementLocated()**

- **alert()**
- **getText() - Alert**
- **accept()**
- **dismiss()**
- **alertIsPresent()**
- **sendKeys()- Alert**
- **moveToElement()**
- **click() - Actions**
- **dragAndDropBy()**
- **contextClick()**
- **doubleClick()**
- **dragAndDrop()**
- **keyDown()**
- **keyUp()**
- **sendKeys() - Actions**
- **chord()**
- **build()**
- **perform()**
- **executeScript()**
- **executeAsyncScript()**
- **getCookie()**
- **getCookieNamed()**
- **addCookie()**
- > **addCookie()**
- > **deleteCookie()**
- > **deleteCookieNamed()**
- > **deleteAllCookies()**
- > **below()**
- > **above()**
- > **toRightOf()**
- > **toLeftOf()**
- > **near()**

## Manually Creating WebDriver Project

1. Open Eclipse IDE and Create New Project as WebDriverProject
2. Download latest Stable selenium JAR files (selenium.dev). Extract it.  
Browser Specific drivers available with JAR ( Stable : 04.18.01)
3. Attach Jar to project  
Right click > Properties > Java build path > libraries > Classpath >  
Add External jars > select all jars > Open > again > Add External jars >  
select all jars inside lib folder > Open > Apply and close.  
you will see Referenced Libraries.

## Create Maven Project ---> Recommended

1. Open Eclipse IDE and Create maven project.
2. Group ID and Artifact ID - project name. click finish
3. open pom.xml and necessary dependencies. (mvnrepository.com)  
- Selenium Java + Web Drivers