

Selenium Locators

Locating by ID ^{Refer Part2-Page37} `driver.findElement(By.id("q")).sendKeys("Selenium 3");`

Locating by Name ^{Refer Part2-Page37} `driver.findElement(By.name("q")).sendKeys("Selenium 3");`

Locating by Xpath ^{Refer Part2-Page37/38} `driver.findElement(By.xpath("//input[@id='q']")).sendKeys("Selenium 3");`

Locating Hyperlinks by Link Text ^{Refer Part2-Page38/39} `driver.FindElement(By.LinkText("edit this page")).Click();`

Locating by DOM ^{Refer Part2-Page39} `dom =document.getElementById('signInForm')`

Locating by CSS ^{Refer Part2-Page40-43} `driver.FindElement(By.CssSelector("#rightbar > .menu > li:nth-of-type(2) > h4"));`

Locating by ClassName ^{Refer Part2-Page37} `driver.findElement(By.className("profile-header"));`

Locating by TagName ^{Refer Part2-Page37} `driver.findElement(By.tagName("select")).Click();`

Locating by LinkText ^{Refer Part2-Page37-40} `driver.findElement(By.LinkText("NextPage")).click();`

Locating by PartialLinkText ^{Refer Part2-Page37-40} `driver.findElement(By.partialLinkText("NextP")).click();`

Handle multiple windows and tabs

`getWindowHandle()` used to retrieve the handle of the current page (a unique identifier)

`getWindowHandles()` used to retrieve a set of handles of the all the pages available

`driver.switchTo().window("windowName/handle")` switch to a window

`driver.close()` closes the current browser window

`driver.quit();` This method Closes all windows opened by the WebDriver.

Switch Frames

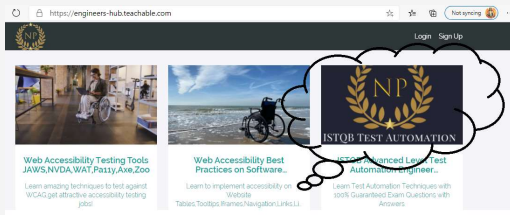
Mentioning the frame index number, the Driver will switch to that specific frame `driver.switchTo.frame(int frameNumber)`

Mentioning the frame element or ID, the Driver will switch to that specific frame `driver.switchTo.frame(string frameNameOrID)`

Mentioning the frame web element, the Driver will switch to that specific frame `driver.switchTo.frame(WebElement frameElement)`

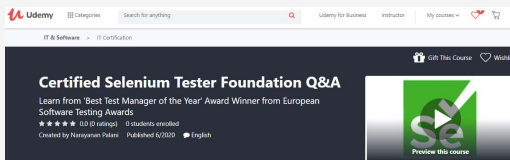
Switching back to the main window `driver.switchTo().defaultContent()`

ISTQB Advanced Test Automation Engineering



<https://engineers-hub.teachable.com/p/istqb-advanced-level-test-automation-engineer-professional-training-with-q-a>

Selenium Q&A



<https://www.udemy.com/course/certified-selenium-tester-foundation/>

Find element VS Find elements

- | | |
|------------------------------|---|
| driver.findElement() | When no match has found(0) throws NoSuchElementException when 1 match found returns a WebElement instance when 2+ matches found returns only the first matching web element |
| driver.findElements() | when no match has found (0) returns an empty list when 1 match found returns a list with one WebElement when 2+ matches found returns a list with all matching WebElements |

Materials

Selenium Certification <https://isqi.org/en/54-certified-selenium-tester-foundation.html>

Selenium Certification Q&A <https://engineers-hub.teachable.com/>

Part1-Software Automation Testing Secrets Revealed <https://www.amazon.com/Software-Automation-Testing--Secrets-Revealed/dp/1729432506/>

Part2-Software Automation Testing Secrets Revealed <https://www.amazon.com/gp/product/1729432379/>

Part3-Software Automation Testing Secrets Revealed <https://www.amazon.com/Advanced-Selenium-Web-Accessibility-Testing/dp/1949449432/>

Udemy Course-Selenium <https://engineers-hub.teachable.com/>

Udemy Course-ISTQB Test Automation Engineering <https://engineers-hub.teachable.com/>

Useful Commands (Credits: Madhan K) <https://medium.com/@madhankumaravelu93/selenium--cheat-sheet-a-comprehensive-list-of-selenium-commands-fa4c5c9d11ab>

Useful Commands (Credits: Bijan Patel) <https://qascript.com/selenium-webdriver-commands-cheat-sheet/>

Handle alerts: (Web-based alert pop-ups)

driver.switchTO().alert.getText()	to retrieve the alert message
driver.switchTO().alert.accept()	to accept the alert box
driver.switchTO().alert.dismiss()	to cancel the alert box
driver.switchTO().alert.sendKeys("Text")	to send data to the alert box

Page Loading Strategy

normal	<pre>ChromeOptions chromeOptions = new ChromeOptions(); chromeOptions.setPageLoadStrategy(PageLoadStrategy.NORMAL); WebDriver driver = new ChromeDriver(chromeOptions);</pre>
eager: When setting to eager, Selenium WebDriver waits until DOMContentLoaded event fire is returned.	<pre>ChromeOptions chromeOptions = new ChromeOptions(); chromeOptions.setPageLoadStrategy(PageLoadStrategy.EAGER); WebDriver driver = new ChromeDriver(chromeOptions);</pre>
none: When set to none Selenium WebDriver only waits until the initial page is downloaded.	<pre>ChromeOptions chromeOptions = new ChromeOptions(); chromeOptions.setPageLoadStrategy(PageLoadStrategy.NONE); WebDriver driver = new ChromeDriver(chromeOptions);</pre>

The document.readyState property of a document describes the loading state of the current document. By default, WebDriver will hold off on responding to a driver.get() (or) driver.navigate().to() call until the document ready state is complete. By default, when Selenium WebDriver loads a page, it follows the normal pageLoadStrategy.

Get methods in Selenium

getTitle()	used to retrieve the current title of the webpage
getCurrentUrl()	used to retrieve the current URL of the webpage
getPageSource()	used to retrieve the current page source of the webpage
getText()	used to retrieve the text of the specified web element
getAttribute()	used to retrieve the value specified in the attribute

Navigation

Refresh Page	driver.navigate().refresh()
Navigate forwards in the browser history	driver.navigate().forward()
Navigate backward in the browser history	driver.navigate().back()

Desired Capabilities-Chrome

```
DesiredCapabilities caps = new DesiredCapabilities();
caps.setCapability("browserName", "chrome");
caps.setCapability("browserVersion", "80.0");
caps.setCapability("platformName", "win10");
WebDriver driver = new ChromeDriver(caps); // Pass the capabilities as an argument to the driver object
```

Desired Capabilities-Firefox

```
DesiredCapabilities caps = new DesiredCapabilities();
caps.setCapability("browserName", "firefox");
caps.setCapability("browserVersion", "81.0");
caps.setCapability("platformName", "win10");
WebDriver driver = new FirefoxDriver(caps); // Pass the capabilities as an argument to the driver object
```

Cookies

addCookie(arg)	<code>driver.manage().addCookie(new Cookie("foo", "bar"));</code>
getCookies()	<code>driver.manage().getCookies();</code> // to get all cookies
getCookieNamed()	<code>driver.manage().getCookieNamed("foo");</code>
deleteCookieNamed()	<code>driver.manage().deleteCookieNamed("foo");</code>
deleteCookie()	<code>Cookie cookie1 = new Cookie("test2", "cookie2"); driver.manage().addCookie(cookie1); driver.manage().deleteCookie(cookie1);</code> // deleting cookie object
deleteAllCookies()	<code>driver.manage().deleteAllCookies();</code> // deletes all cookies

Take screenshot

Capture the screenshot and store	<code>getScreenshotAs</code>
----------------------------------	------------------------------

`getScreenshotAs` — used to Capture the screenshot and store it in the specified location. This method throws `WebDriverException`. `copy()` method from the `File Handler` class is used to store the screenshot in a destination folder.

Code Example:

```
TakesScreenshot screenShot =(TakesScreenshot)-
driver;
FileHandler.copy(screenShot.getScreenshotAs(OutputType.FILE), new File("path/to/destination/folder/screenshot.png"));
```

Practice Scripts

Execute Javascript

Executes an asynchronous piece of JavaScript `executeAsyncScript()`

Executes JavaScript `executeScript()`

Code Example:

```
if (driver instanceof JavascriptExecutor) {
    ((JavascriptExecutor)driver).executeScript("alert-
('hello world');");
}
```

Annotations

TestNG `@BeforeSuite @AfterSuite @BeforeTest @AfterTest @BeforeGroups @AfterGroups @BeforeClass @AfterClass @BeforeMethod @AfterMethod`

JUnit `@After @AfterClass @Before @BeforeClass @Ignore @Test`

Selenium Exceptions

'Server Exception : sessionId should not be null' Refer Part2--Page160 Replace Chrome Driver with Latest Version

ERROR: No sessionId provided Refer Part2--Page160 Use Jenkins

UnreachableBrowserException Refer Part2--Page160 Registry Key Changes Recommended

Illegal state exception using Selenium 3 Refer Part2--Page161 Get latest Firefox Gecko Driver

Launch

Webpage Refer Part2--Page60

```
driver.get("www.webdriver-
inselenium.com");
```

Click Button

Refer Part2--Page60

```
driver.findElement(By.id("su-
bmit")).click();
```

Store Text

Refer Part2--Page60

```
String txtDropdown =
driver.findElement(By.ID("se-
lect")).getText();
```

Enter Text

Refer Part2--Page61

```
driver.findElement(By.xpath-
("//input[@name='FirstNam-
e034']")).sendKeys("Naraya-
nanPalani");
```

Handle Alert

Refer Part2--Page62

```
Alert Alertpopup = driver.sw-
itchTo().alert();
```

(Mouse)Click

Refer Part2--Page62

```
driver.findElement(By.xpath-
("//input[@value='OBJECT
NAME']")).click();
```

Compare Text

Refer Part2--Page64

```
Assert.assertTrue(chkbox-
12.isSelected());
```

Disable a Field

Refer Part2--Page64

```
driver.getElementsByName-
('<ObjectID>')[0].setAttrib-
ute('disabled', "");
```

Enable a Field

Refer Part2--Page64

```
driver.getElementsByName-
('<ObjectID>')[0].removeAtt-
ribute('disabled');
```

Screenshot

Refer Part2--Page64

```
File snapshot = ((TakesScre-
enshot)driver).getScreenshot-
As(OutputType.FILE);
```

```
FileUtils.copyFile(snapshot,
new File("C:\\screenshot.j-
pg"));
```

Print the Title of the Page

Refer Part2--Page68

```
String pagetitle = driver.ge-
tTitle();
```

```
System.out.print(pagetitle);
```

Implicit Wait

Refer Part2--Page69

```
driver.manage().timeouts().i-
mplicitlyWait(10, TimeUnit.S-
ECONDS);
```

Explicit Wait

Refer Part2--Page70

```
WebDriverWait wait = new
WebDriverWait(driver, 20);
```

```
wait.until(ExpectedConditions-
.textToBePresentInElem-
```