**TABLE OF CONTENTS**Sample Selenium Script

|  |
| --- |
| a) Locators - Name,tagName, ClassName, linkText, partialLinkText, cssSelector, xpath, ID  b) Checkbox Handling  c) Refreshing the browser  d) Capturing Screenshot |

**A Drop down handling**

|  |
| --- |
| <html><body><select id = "designation">  <option value = "MD">MD</option>  <option value = "prog"> Programmer </option>  <option value = "CEO"> CEO </option>  </option></select><body>  </html> |

|  |
| --- |
| Select dropdown = new Select(driver.findElement(By.id("identifier")));  dropdown.selectByValue("prog");  dropdown.selectByVisibleText("Programmer ");  dropdown.selectByIndex(1); |

1) Switching to frame driver.switchTo.frame("Frame\_ID");driver.switchTo().defaultContent();

2) Computer Record UI Record Stale Element Handling Property File Extraction 3) Certificate and Profile Handling firefox and chrome launching

4) Wait until element visible

**WINDOW HANDLES**

**Switching back to parent window handles**

//Stores the handle details String windowHandle=driver.getWindowHandles();

//Alert portion Alert alert = driver.switchTo().alert(); alert.accept();

//After switching back to main window.. driver.navigate.windows(windowHandle);

**Switching to Windows Handle and performing operations**

String parentHandle = driver.getWindowHandle();

// get the current window handledriver.findElement(By.xpath("//\*[@id='someXpath']")).click();

// click some link that opens a new window

for (String winHandle : driver.getWindowHandles()) { driver.switchTo().window(winHandle);

// switch focus of WebDriver to the next found window handle (that's your newly opened window)}

//code to do something on new windowdriver.close();

// close newly opened window when done with itdriver.switchTo().window(parentHandle);

// switch back to the original window

**SELENIUM GRID AND NODE**

**Starting Node**

java -jar selenium-server-standalone-2.46.0.jar -role node -Dwebdriver.ie.driver=IEDriverServer.exe -hub <http://localhost:4444/grid/register> -browser browserName="internet explorer",version=11,maxInstances=5,platform=WINDOWS -browser browserName=firefox,version=40.0.3,maxInstances=5,platform=WINDOWSjava –jar selenium-server-standalone-2.30.0.jar –role webdriver –hub <http://192.168.1.3:4444/grid> /register -port 5566**Starting Hub**java -jar selenium-server-standalone-2.46.0.jar -role hub

**Launch browser method calling the node from the code**

|  |
| --- |
| public void launchbrowser() throws FormedURLException { DOMConfigurator.configure("src/log4j.xml"); log.info("Launching the Internet Explorer"); System.setProperty("webdriver.ie.driver", "lib/IEDriverServer.exe");DesiredCapabilities capability = DesiredCapabilities.internetExplorer(); driver = new RemoteWebDriver(new URL("http://114.9.226.35:4444/wd/hub"), capability);//driver = new InternetExplorerDriver(); log.info("Internet Explorer Launched");//driver = new FirefoxDriver(); log.info("Launching the Application URL");driver.get(Constant.URL);} |

public void setup() throws FormedURLException { baseUrl = “http://newtours.demoaut.com”; nodeUrl = “http://192.168.1.4:5566/wd/hub”; DesiredCapabilities capability = DesiredCapabilities.firefox(); capability.setBrowserName(“firefox”); capability.setPlatform(Platform.XP); driver = new RemoteWebDriver(new URL(nodeURL), capability);**}**

**ROBOT FRAMEWORK**

3) Hit Enter. WebElement link = driver.findElement(By.xpath("myxpath"));clickAndSaveFileIE(link);public static void clickAndSaveFileIE(WebElement element) throws InterruptedException{ try { Robot robot = new Robot(); //get the focus on the element..don't use click since it stalls the driver element.sendKeys(""); //simulate pressing enter robot.keyPress(KeyEvent.VK\_ENTER); robot.keyRelease(KeyEvent.VK\_ENTER); //wait for the modal dialog to open Thread.sleep(2000); //press s key to save robot.keyPress(KeyEvent.VK\_S); robot.keyRelease(KeyEvent.VK\_S); Thread.sleep(2000); //press enter to save the file with default name and in default location robot.keyPress(KeyEvent.VK\_ENTER); robot.keyRelease(KeyEvent.VK\_ENTER); } catch (AWTException e) { e.printStackTrace(); }

2) Write a TestNG test - Open a website in chrome and firefox, read a data sets from excel sheet. Go to the list box and multi select the values found in excel sheet. Refresh the page. Accept and close an alert that pops up. Take screenshot of the alert.

**Opening in Chrome and Firefox with two different drivers:-**

**public** **void** launchIEbrowser() **throws** FormedURLException { log.info("Launching the Internet Explorer"); System.*setProperty*("webdriver.ie.driver", "lib/IEDriverServer.exe"); DesiredCapabilities capability = DesiredCapabilities.internetExplorer(); driver = **new** RemoteWebDriver(**new** URL("http://114.9.226.35:4444/wd/hub"), capability); log.info("Internet Explorer Launched"); log.info("Launching the Application URL"); driver.get(Constant.URL); }**public** **void** launchFirefoxBrowser() **throws** MalformedURLException { log.info("Launching the Firefox browser"); DesiredCapabilities capability = DesiredCapabilities.firefox(); capability.setBrowserName(“firefox”); driver = **new** RemoteWebDriver(**new** URL("http://114.9.226.35:4444/wd/hub"), capability); log.info("Firefox browser Launched"); log.info("Launching the Application URL"); driver.get(Constant.URL); }

**Read Data from Excel Sheet**

String FilePath = "D:\\sample\_data.xls";FileInputStream fs = **new** FileInputStream(FilePath);Workbook wb = Workbook.getWorkbook(fs);Sheet sh1 = wb.getSheet(0); String listChoice = sh1.getCell(1,j).getContents();

**Selecting the particular list choice**

Select listbox = **new** Select(driver.findElement(By.xpath("//select[@name='FromLB']")));listbox.selectByVisibleText(listChoice);

**Refresh the browser**

driver.navigate().refresh();

**Accept and close an alert that pops up**

Alert A1 = driver.switchTo().alert();//To read the text from alert popup.

**Taking screenshot of alert**

File screenshot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);FileUtils.copyFile(screenshot, **new** File("D:\\screenshot.jpg"));

**Close the alert and close the driver**

A1.accept();A1.close();driver.quit();

**1) Launch Browser**

**public** **void** launchbrowser(){ System.*setProperty*("webdriver.ie.driver", "C:\\IEDriver\\IEDriverServer.exe"); *driver* = **new** InternetExplorerDriver(); //driver = new FirefoxDriver(); *driver*.get(Constant.***URL***); }

// Launching the Browser

cmb.launchbrowser(); **try**{ cmb.maximize(); } **catch**(Throwable e){ System.***out***.println("no maximixe"); }

**2) Close Browser**

**public** **void** closeBrowser(){ *driver*.close();}

3) **Maximize**

**public** **void** maximize(){*driver*.manage().window().maximize();}

**4) Alert**

**public** **void** alert(){**try**{ Alert alert = *driver*.switchTo().alert(); alert.accept(); System.***out***.println("handled alert"); } **catch** (Throwable e){ System.***out***.println("no alert"); } }**try**{ cmb.alert(); System.***out***.println("handled alert"); } **catch** (Throwable e){ System.***out***.println("no alert"); }

5) **Back**

**public** **void** back(){ *driver*.navigate().back();}

**6) Click Element**

**public** **void** clickElement(String clkObject){ **try** { *driver*.findElement(By.*xpath*(clkObject)).click(); // driver.findElement(By.xpath(utility.getProperty("Click\_SignIn"))).click(); System.***out***.println("Clicked" +clkObject+"Succesfully"); *logger*.info("Clicked\_by\_xpath"+ clkObject +"' clicked successfully"); } **catch**(Exception e){ System.***out***.println("Clicked\_by\_xpath" +clkObject+ "Passed"); }}

//Clicking on the SignIn Link

cmb.clickElement(utility.getProperty("Click\_SignIn"));Thread.*sleep*(2000);

//Switching to New Frame

cmb.SwitchFrames("DOMWindowIframe152");

**7) Switch Frame**

**public** **void** SwitchFrames(String FrameName){*driver*.switchTo().frame(FrameName); *driver*.manage().timeouts().implicitlyWait(10, TimeUnit.***SECONDS***);}

**8) Window Handle**

**public** **void** windowhandle(){//To get handle of current window // Switch to new window opened String parentWindow = *driver*.getWindowHandle(); **for**(String winHandle : *driver*.getWindowHandles()){ *driver*.switchTo().window(winHandle); }

// Perform the actions on new window // Close the new window, if that window no more required *driver*.close(); // Switch back to original browser (first window) *driver*.switchTo().window(parentWindow);}

**Robot Class Implementation**

1) Click link or Press Enter key on the link.

2) type S.

3) Hit Enter.

WebElement link = driver.findElement(By.xpath("myxpath"));clickAndSaveFileIE(link);**public** **static** **void** clickAndSaveFileIE(WebElement element) **throws** InterruptedException{**try** {Robot robot = **new** Robot();//get the focus on the element..don't use click since it stalls the driverelement.sendKeys("");//simulate pressing enterrobot.keyPress(KeyEvent.VK\_ENTER);robot.keyRelease(KeyEvent.VK\_ENTER);//wait for the modal dialog to openThread.sleep(2000);

//press s key to save

robot.keyPress(KeyEvent.VK\_S);robot.keyRelease(KeyEvent.VK\_S);Thread.sleep(2000);

//press enter to save the file with default name and in default location

robot.keyPress(KeyEvent.VK\_ENTER);robot.keyRelease(KeyEvent.VK\_ENTER);} **catch** (AWTException e) {e.printStackTrace();}

* **Sample Selenium Script**

public class SeleniumWebDriverSample {public static void main(String[] args) { // Create a new instance of the Firefox driver WebDriver driver = new FirefoxDriver(); // And now use this to visit Google driver.get("http://www.google.com"); // Find the text input element by its name WebElement element = driver.findElement(By.name("q")); // Enter something to search for element.sendKeys("Cheese!"); // Now submit the form. element.submit(); // Check the title of the page System.out.println("Page title is: " + driver.getTitle()); // Should see: "cheese! - Google Search" System.out.println("Page title is: " + driver.getTitle()); //Close the browser driver.quit(); }}

**1) Eight types of Locators using findElement**

1)**By ClassName:-**<div class="cheese"><span>Cheddar</span></div><div class="cheese"><span>Gouda</span></div> List<WebElement> elements = driver.findElements(By.className("cheese"));

2)**By TagName:-**<iframe src="..."></iframe>

WebElement frame = driver.findElement(By.tagName("iframe"));

3)**By name** <input name="cheese" type="text"/>

WebElement element = driver.findElement(By.name("cheese"));

4)**By Link text** <a href="http://www.google.com/search?q=cheese">cheese</a>>

WebElement element = driver.findElement(By.linkText("cheese"));

5)**By Partial Link text** <a href="http://www.google.com/search?q=cheese">search for cheese</a>>

WebElement element = driver.findElement(By.partialLinkText("cheese"));

6)**By Css Selector** <div id="food"><span class="dairy">milk</span><span class="dairy aged">cheese</span></div>

WebElement element = driver.findElement(By.cssSelector("#food span.dairy.aged"));

7)**By Xpath** <input type="text" name="example" />

List<WebElement> element = driver.findElements(By.xpath("//input"));

8)**By Id** <div id="coolestWidgetEvah">...</div>

WebElement element = driver.findElement(By.id("coolestWidgetEvah"));

**2) Handling Checkboxes**

WebElement checkboxTable = driver.findElement(By.class("table5"));List<WebElement> allCheckboxOptions = checkboxTable.findElements(By.cssSelector(".table5>input")); **for** (WebElement singleCheckboxOption : allCheckboxOptions) { // Print the checkbox option and click it System.out.println(singleCheckboxOption.getAttribute("value")); singleCheckboxOption.click();}

**3) Refreshing the browser**

**4) Webdriver – captureEntirePageScreenshot**

File screenshot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);FileUtils.copyFile(screenshot, new File("D:\\screenshot.jpg"));@Test public void test () throws InterruptedException, IOException { **//Capture entire page screenshot and then store it to destination drive** File screenshot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE); FileUtils.copyFile(screenshot, new File("D:\\screenshot.jpg")); System.out.print("Screenshot is captured and stored in your D: Drive"); }

**5) Handling Alerts**

public class unexpected\_alert { WebDriver driver; @BeforeTest public void setup() throws Exception { driver =new FirefoxDriver(); driver.manage().window().maximize(); driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS); driver.get("http://only-testing-blog.blogspot.in/2014/01/textbox.html"); } @AfterTest public void tearDown() throws Exception { driver.quit(); } @Test public void Text() throws InterruptedException {

// Type1 – Alert

**// Locate the alert pop up by clicking a button**

driver.findElement(By.xpath("//input[@value='Show Me Alert']")).click();

**// Switch to alert pop up**

Alert A1 = driver.switchTo().alert();

**//To read the text from alert popup.**

String Alert1 = A1.getText(); System.out.println(Alert1); Thread.sleep(2000);

**//To accept/Click Ok on alert popup.k**

A1.accept();

**//Prompt Pop up Handling**

driver.findElement(By.xpath("//button[contains(.,'Show Me Prompt')]")).click(); Alert A3 = driver.switchTo().alert(); String Alert3 = A3.getText(); System.out.println(Alert3);

**//To type text In text box of prompt pop up.**

A3.sendKeys("This Is John"); Thread.sleep(2000); A3.accept(); }}

* **Handling Forms**

package Testng\_Pack; import java.util.concurrent.TimeUnit; import org.openqa.selenium.By;import org.openqa.selenium.WebDriver;import org.openqa.selenium.firefox.FirefoxDriver;import org.testng.annotations.AfterTest;import org.testng.annotations.BeforeTest;iimport org.testng.annotations.Test; public class Form\_Submit { WebDriver driver = new FirefoxDriver(); @BeforeTest public void setup() throws Exception { driver.manage().window().maximize(); driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS); driver.get("http://only-testing-blog.blogspot.in/2014/05/form.html"); } @AfterTest public void tearDown() throws Exception { driver.quit(); } @Test public void LogIn\_Test(){ driver.findElement(By.xpath("//input[@name='FirstName']")).sendKeys("MyFName"); driver.findElement(By.xpath("//input[@name='LastName']")).sendKeys("MyLName"); driver.findElement(By.xpath("//input[@name='EmailID']")).sendKeys("My Email ID"); driver.findElement(By.xpath("//input[@name='MobNo']")).sendKeys("My Mob No."); driver.findElement(By.xpath("//input[@name='Company']")).sendKeys("My Comp Name");**//To submit form. //You can use any other Input field's(First Name, Last Name etc.) xpath too In bellow given syntax.** driver.findElement(By.xpath("//input[@name='Company']")).submit(); String alrt = driver.switchTo().alert().getText(); driver.switchTo().alert().accept(); System.out.println(alrt); }}

### **Locating Web Element By ClassName In Selenium WebDriver with example**

You can get the class name of element using firebug as shown in below given image.

**driver.findElement(By.className("date-header"));**

package junitreportpackage; import java.util.concurrent.TimeUnit;import org.junit.After;import org.junit.Before;import org.junit.Test;import org.openqa.selenium.By;import org.openqa.selenium.WebDriver;import org.openqa.selenium.WebElement;import org.openqa.selenium.firefox.FirefoxDriver; public class Mytest1 { **//To open Firefox browser** WebDriver driver = new FirefoxDriver(); @Before public void beforetest() { **//To Maximize Browser Window** driver.manage().window().maximize(); **//To Open URL In browser** driver.get("http://only-testing-blog.blogspot.in/2013/11/new-test.html"); } @After public void aftertest() { driver.quit(); } @Test public void test() { driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS); String datentime = driver.findElement(By.className("date-header")).getText();//Locating element by className and store its text to variable datentime. System.out.print(datentime); } }

### How To Locate Elements By ID In Selenium WebDriver With Example

**Locating UI Element By ID**

**driver.findElement(By.id("submitButton"));**@Test public void test() { driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS); for (int i = 0; i<=20; i++) { WebElement btn = driver.findElement(By.id("submitButton"));//Locating element by id if (btn.isEnabled()) { //if webelement's attribute found enabled then this code will be executed. System.out.print("\nCongr8s... Button is enabled and webdriver is clicking on it now"); //Locating button by id and then clicking on it. driver.findElement(By.id("submitButton")).click(); i=20; } else { //if webelement's attribute found disabled then this code will be executed. System.out.print("\nSorry but Button is disabled right now.."); } try { Thread.sleep(500); } catch (InterruptedException e) { e.printStackTrace(); } } }

### Evaluating javascript in selenium webdriver to get page title with example

**JavascriptExecutor** is very useful Interface in **webdriver**. Interface JavascriptExecutor helps you to execute javascript in your test case whenever required. If you knows/remember, We had seen [**Different examples of evaluating javascript in selenium IDE**](http://software-testing-tutorials-automation.blogspot.in/search/label/Using%20javascript%20with%20selenium%20IDE)to perform different actions. Let me you one example of

evaluating javascript in selenium WebDriver.

Sometimes in your test case, you needs to store your software web application **page title** to compare it with expected page title. In selenium IDE, we can use "[**storeTitle**](http://software-testing-tutorials-automation.blogspot.in/2012/11/storelocation-and-storetitle-commands.html)" command to store it in variable. In webdriver ,we can do it directly using **driver.getTitle();** as shown in [**this example**](http://software-testing-tutorials-automation.blogspot.in/2014/01/selenium-webdriver-wait-for-title-with.html). But if you wants to do it using javascript then how will you do it?

**Example : Get page title using javascript in selenium webdriver**

Copy bellow given @Test method part of get page title using javascript example and replace it with the @Test method part of example given on [**this page**](http://software-testing-tutorials-automation.blogspot.in/2014/01/how-to-wait-for-element-to-be-clickable.html)**.** (Note : @Test method is marked with **pink color in** that linked page).

@Test public void test () { JavascriptExecutor javascript = (JavascriptExecutor) driver; **//Get current page title** String pagetitle=(String)javascript.executeScript("return document.title"); System.out.println("My Page Title Is : "+pagetitle); **//Get current page URL** String CurrentURL = driver.getCurrentUrl(); System.out.println("My Current URL Is : "+CurrentURL); }

### Selenium WebDriver : Verify Element Present In Selenium WebDriver

Some times you need to verify the presence of element before taking some action on software web application page. As you know, [**Selenium IDE**](http://software-testing-tutorials-automation.blogspot.in/2013/07/list-of-selenium-commands-with-examples.html) has many built in commands to perform different types of actions on your software web application page. You can verify presence of element by using  
 "[**verifyElementPresent**](http://software-testing-tutorials-automation.blogspot.in/2013/06/using-selenium-verifyelementpresent-and.html)" command in selenium IDE. Also you can view example of selenium IDE "[**verifyElementNotPresent**](http://software-testing-tutorials-automation.blogspot.in/2014/02/selenium-ide-verifyelementnotpresent.html)" command. Web driver have not any built in method or interface by which we can verify presence of element on the page.

Yes we can do it very easily in WebDriver too using bellow given syntax.

Boolean iselementpresent = driver.findElements(By.xpath("//input[@id='text2']")).size()!= 0;

We have to use findElements() method for this purpose. Above syntax will return true if element is present on page. Else it will return false. You can put if condition to take action based on presence of element.

Bellow given example will check the presence of different text box on page. It will print message in console based on presence of element.

Copy bellow given @Test method part of iselementpresent example and replace it with the @Test method part of example given on [**THIS PAGE**](http://software-testing-tutorials-automation.blogspot.in/2014/02/selenium-webdriver-tutorial-how-to.html)**.** (Note : @Test method is marked with **pink color in** that linked page).

@Test public void test () throws InterruptedException { for (int i=1; i<6; i++) { **//To verify element is present on page or not.** String XPath = "//input[@id='text"+i+"']"; Boolean iselementpresent = driver.findElements(By.xpath(XPath)).size()!= 0; if (iselementpresent == true) { System.out.print("\nTargeted TextBox"+i+" Is Present On The Page"); } else { System.out.print("\nTargeted Text Box"+i+" Is Not Present On The Page"); } } }How To Deselect Option By Visible Text Or Value Or By Index In Selenium WebDriver With Example

If below method is appearing strange and overloaded, then use previous method of for looping list box contents and individually clicking checkboxes.  
 Sample Test exemplifying Select Listbox functionality.

@Test public void test () throws InterruptedException { driver.findElement(By.id("text1")).sendKeys("My First Name"); Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']"))); listbox.selectByVisibleText("USA"); listbox.selectByVisibleText("India"); Thread.sleep(1000); **//To deselect by visible text** listbox.deselectByVisibleText("India3"); Thread.sleep(1000); listbox.selectByValue("India5"); listbox.selectByValue("India7"); Thread.sleep(1000); **//To deselect by value** listbox.deselectByValue("Mexico"); Thread.sleep(1000); listbox.selectByIndex(4); listbox.selectByIndex(5); Thread.sleep(1000); **//To deselect by index** listbox.deselectByIndex(5); Thread.sleep(1000); driver.findElement(By.xpath("//input[@value='->']")).click(); Thread.sleep(1000); } }

### How To Select Dropdown Value With Example

In bellow given example, First imported webdriver package "org.openqa.selenium.support.ui.Select" to get support of webdriver class "Select".

import org.openqa.selenium.support.ui.Select; Now we can use "Select" class to identify drop down from web page by writing bellow given syntax.Select mydrpdwn = new Select(driver.findElement(By.id("Carlist"))); Now we can select any value from selected drop down as bellow.mydrpdwn.selectByVisibleText("Audi"); Full Example to select value from drop down is as bellow. **EXAMPLE**package junitreportpackage; import java.util.concurrent.TimeUnit; import org.junit.After;import org.junit.Before;import org.junit.Test;import org.openqa.selenium.By;import org.openqa.selenium.JavascriptExecutor;import org.openqa.selenium.WebDriver;import org.openqa.selenium.WebElement;import org.openqa.selenium.firefox.FirefoxDriver;import org.openqa.selenium.support.ui.ExpectedConditions;import org.openqa.selenium.support.ui.Select;import org.openqa.selenium.support.ui.WebDriverWait; public class Mytesting { WebDriver driver = new FirefoxDriver(); @Before public void beforetest() { driver.manage().window().maximize(); driver.get("http://only-testing-blog.blogspot.in/2014/01/textbox.html"); } @After public void aftertest() { driver.quit(); } // using previous methodology involving lists and for loop. @Test public void test () throwsInterruptedException { driver.findElement(By.id(“text1”)).sendKeys(“My First Name”); WebElement table5 = driver.findElement(By.class(“LOCATOR1”)); List<WebElement> dropboxOptions = driver.findElement(By.cssSelector(“.table5 > input”)); for (WebElement singleDropboxOption: dropboxOptions) { if (singleDropboxOption.getAttribute(“value”)==”Yogurt”) singleDropboxOption.click(); } } @Test public void test () throws InterruptedException { driver.findElement(By.id("text1")).sendKeys("My First Name"); **//Selecting value from drop down using visible text** Select mydrpdwn = new Select(driver.findElement(By.id("Carlist"))); mydrpdwn.selectByVisibleText("Audi"); WebDriverWait wait = new WebDriverWait(driver, 15); wait.until(ExpectedConditions.elementToBeClickable(By.id("text2"))); } }

## Accessing Lnks & Tables using Selenium Webdriver

In this tutorial, we are going to learn about accessing links & Tables using Webdriver

## Accessing Links

### Links Matching a Criterion

Links can be accessed using an exact or partial match of their link text. The examples below provide scenarios where multiple matches would exist, and would explain how WebDriver would deal with them.

### Exact Match

**Accessing links using their exact link text is done through the By.linkText() method**. However, if there are two links that have the very same link text, this method will only access the first one. Consider the HTML code below

When you try to run the WebDriver code below, you will be accessing the first "click here" link

As a result, you will automatically be taken to Google.

### Partial Match

Accessing links using a portion of their link text is done using the **By.partialLinkText()** method. If you specify a partial link text that has multiple matches, only the first match will be accessed. Consider the HTML code below.

When you execute the WebDriver code below, you will still be taken to Google.

### Case-sensitivity

The parameters for **By.linkText()** and **By.partialLinkText()** are both case-sensitive, meaning that capitalization matters. For example, in Mercury Tours' homepage, there are two links that contain the text "egis" - one is the "REGISTER" link found at the top menu, and the other is the "Register here" link found at the lower right portion of the page.

Though both links contain the character sequence "egis", the "By.partialLinkText()" method will access these two links separately depending on capitilization of the characters. See the sample code below.

The output above confirms that both links were accessed successfully because their respective page titles were retrieved correctly.

### Accessing Image Links

Image links are images that act as references to other sites or sections within the same page. Since they are images, we cannot use the By.linkText() and By.partialLinkText() methods because image links basically have no link texts at all. In this case, we should resort to using either By.cssSelector or By.xpath. The first method is more preferred because of its simplicity.

In the example below, we will access the "Facebook" logo on the upper left portion of Facebook's Password Recovery page.

We will use By.cssSelector and the element's "title" attribute to access the image link. And then we will verify if we are taken to Facebook's homepage.

## Reading a Table

There are times when we need to access elements (usually texts) that are within HTML tables. However, it is very seldom for a web designer to provide an id or name attribute to a certain cell in the table. Therefore, we cannot use the usual methods such as "By.id()", "By.name()", or "By.cssSelector()". In this case, the most reliable option is to access them using the "By.xpath()" method

### XPath Syntax

We will use XPath to get the inner text of the cell containing the text "fourth cell".

## Accessing Forms using Selenium WebDriver

In this tutorial , we will learn how to access forms and its elements using Webdriver

## Accessing Form Elements

**Input Box**

Input boxes refer to either of these two types:

1. **Text Fields**- text boxes that accept typed values and show them as they are.
2. **Password Fields**- text boxes that accept typed values but mask them as a series of special characters (commonly dots and asterisks) to avoid sensitive values to be displayed.

### Entering Values in Input Boxes

The **sendKeys()** method is used to enter values into input boxes.

### Deleting Values in Input Boxes

The **clear()** method is used to delete the text in an input box. **This method does not need any parameter**. The code snippet below will clear out the text "tutorial" in the User Name text box.

### Radio Button

Toggling a radio button on is done using the **click()** method.

### Check Box

Toggling a check box on/off is also done using the **click()** method.

The code below will click on Facebook's "Keep me logged in" check box twice and then output the result as TRUE when it is toggled on, and FALSE if it is toggled off.

### Links

Links also are accessed by using the **click()** method.

Consider the below link found in Mercury Tours' homepage.

You can access this link using linkText() or partialLinkText() together with click(). Either of the two lines below will be able to access the "Register here" link shown above.

## PDF , Emails and Screenshot of Test Reports in Selenium

Before we look into anything else, lets first understand

### Why do we need reporting ?

When we are using Selenium or any other automation tool, we are performing operations on the web application. But our purpose of automation is not just to exercise the Application Under Test. We, as an automation tester are supposed to test the application, find bugs and report it to development team or higher management. Here the reporting gets importance for software [testing](http://www.guru99.com/software-testing.html) process

REPORTING

### TestNG Reporting

TestNG library provides a very handy reporting feature. After execution, TestNG will generate a test-output folder at the root of the project. This folder contains two type of Reports

**Index.html:** This is the complete report of current execution which contains information like error, groups, time ,reporter logs, testng xml files.

**emailable-report.html:** This is the summarize report of the current test execution which contains test case message in green (for pass test cases) and red(for failed test cases) highlight.

### How to customize TestNG Report

TestNG reporting is quite handy but still sometimes we need some less data in reports or want to display reports in some other format like pdf, excel etc. or want to change report's layout.

There can be two ways we can customize TestNG report

* Using ITestListener Interface:
* Using IReporter Interface:

#### ITestListener Interface

We use this interface when we need to customize real time report. In other words if we are executing bunch of test cases in a TetNG suite and we want to get report of each individual test case, then after each test case we need to implement ITestListener interface. This interface will override onTestFailure ,onTestStart

, onTestSkipped method to send the correct status of the current test case.

Here are the steps we will follow

* Implement methods of iTestListener
* Create test method and add RealGuru99Report class as a listener in Test method class.

**Code Example**

RealGuru99TimeReport.java is the real time reporting class. It will implement ITestListener interface for reporting

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71 | package testNGReport.realTimeReport;  import org.testng.ITestContext;  import org.testng.ITestListener;  import org.testng.ITestResult;  public class RealGuru99TimeReport implements ITestListener{  @Override  public void onStart(ITestContext arg0) {  System.out.println("Start Of Execution(TEST)->"+arg0.getName());  }  @Override  public void onTestStart(ITestResult arg0) {  System.out.println("Test Started->"+arg0.getName());  }    @Override  public void onTestSuccess(ITestResult arg0) {  System.out.println("Test Pass->"+arg0.getName());  }    @Override  public void onTestFailure(ITestResult arg0) {  System.out.println("Test Failed->"+arg0.getName());  }    @Override  public void onTestSkipped(ITestResult arg0) {  System.out.println("Test Skipped->"+arg0.getName());  }    @Override  public void onFinish(ITestContext arg0) {  System.out.println("END Of Execution(TEST)->"+arg0.getName());  }    @Override  public void onTestFailedButWithinSuccessPercentage(ITestResult arg0) {  // TODO Auto-generated method stub  }  } |

What is Cross Browser Testing?

**Cross Browser Testing** is a type of functional test to check that your web application works as expected in different browsers.

### Why do we need Cross Browser Testing?

Web based applications are totally different from windows applications. A web application can be opened in any browser by the end user. For example some people prefer to open [**http://twitter.com**](http://twitter.com/)in **Firefox browser,** while other's can be using **Chrome browser** or **IE**.

In the diagram below you can observe that in **IE**, the login box of twitter is not showing curve at all corners but we are able to see it in chrome browser.

So we need to ensure that the web application will work as expected in all popular browsers, so that more people can access it and use it.

This motive can be fulfilled with Cross Browser Testing of the product.

**Reason Cross Browser Issues**

1. Font size mismatch in different browsers.
2. [JavaScript](http://www.guru99.com/interactive-javascript-tutorials.html) implementation can be different.
3. CSS,HTML validation difference can be there.
4. Some browser still not supporting HTML5.
5. Page alignment and div size.
6. Image orientation.
7. Browser incompatibility with OS. Etc

### How to perform Cross Browser Testing

If we are using Selenium WebDriver, we can automate test cases using Internet Explorer, FireFox, Chrome, Safari browsers.

To evaluate test cases with different browsers in the same machine at same time we can integrate TestNG [framework](http://www.guru99.com/quick-test-professional-qtp-tutorial-34.html) with Selenium WebDriver.

Your testing.xml will look like that,

This testing.xml will map with the test case which will look like that

Here because the testing.xml has two Test tags ('ChromeTest','FirefoxTest'),this test case will execute two times for 2 different browsers.

First Test 'ChromeTest' will pass the value of parameter 'browser' as 'chrome' so ChromeDriver will be executed. This test case will run on Chrome browser.

Second Test 'FirefoxTest' will pass the value of parameter 'browser' as 'firefox' so FirefoxDriver will be executed. This test case will run on FireFox browser.

Complete Code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63 | package parallelTest;  import java.util.concurrent.TimeUnit;  import org.openqa.selenium.By;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.WebElement;  import org.openqa.selenium.chrome.ChromeDriver;  import org.openqa.selenium.firefox.FirefoxDriver;  import org.testng.annotations.BeforeTest;  import org.testng.annotations.Parameters;  import org.testng.annotations.Test;  public class SunilKumarCrossBrowserScript {  WebDriver driver;  /\*\*  \* This function will execute before each Test tag in testng.xml  \* @param browser  \* @throws Exception  \*/  @BeforeTest  @Parameters("browser")  public void setup(String browser) throws Exception{  //Check if parameter passed from TestNG is 'firefox'  if(browser.equalsIgnoreCase("firefox")){  //create firefox instance  driver = new FirefoxDriver();  }    //Check if parameter passed as 'chrome'  else if(browser.equalsIgnoreCase("chrome")){    //set path to chromedriver.exe You may need to download it from <http://code.google.com/p/selenium/wiki/ChromeDriver>  System.setProperty("webdriver.chrome.driver","C:\\chromedriver.exe");  //create chrome instance  driver = new ChromeDriver();  }    else if(browser.equalsIgnoreCase("ie")){  //set path to IEdriver.exe You may need to download it from  // 32 bits <http://selenium-release.storage.googleapis.com/2.42/IEDriverServer_Win32_2.42.0.zip>  // 64 bits <http://selenium-release.storage.googleapis.com/2.42/IEDriverServer_x64_2.42.0.zip>  System.setProperty("webdriver.ie.driver","C:\\IEdriver.exe");  //create chrome instance  driver = new InternetExplorerDriver();  }  else{  //If no browser passed throw exception  throw new Exception("Browser is not correct");  }  driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);  }  @Test  public void testParameterWithXML() throws InterruptedException{  driver.get("http://demo.guru99.com/V4/");  //Find user name  WebElement userName = driver.findElement(By.name("uid"));  //Fill user name  userName.sendKeys("guru99");  //Find password  WebElement password = driver.findElement(By.name("password"));  //Fill password  password.sendKeys("guru99");  }} |

**testing.xml**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE suite SYSTEM "<http://testng.org/testng-1.0.dtd>">  <suite name="TestSuite" thread-count="2" parallel="tests" >  <test name="ChromeTest">  <parameter name="browser" value="Chrome" />  <classes>  <class name="parallelTest.CrossBrowserScript">  </class>  </classes>  </test>  <test name="FirefoxTest">  <parameter name="browser" value="Firefox" />  <classes>  <class name="parallelTest.CrossBrowserScript">  </class>  </classes>  </test>  <test name="IETest">  <parameter name="browser" value="IE" />  <classes>  <class name="parallelTest.CrossBrowserScript">  </class>  </classes>  </test>  </suite> |

**NOTE:** To run the test , Right click on the **testing.xml ,** Select Run As and Click TestNG

Summary

1. Cross browser Testing is a technique to test web application with different web browsers.
2. Selenium can support different type of browsers for automation.
3. Selenium can be integrated with TestNG to perform Cross Browser Testing.
4. From parameters in testing.xml we can pass browser name and in test case we can create WebDriver reference accordingly.

## All About Excel in Selenium: POI & JXL

File IO is a critical part for any software process. We frequently create a file, open it & update something or delete it in our Computers. Same is the case with Selenium Automation. We need a process to manipulate files with Selenium.

Java provides us different classes for File Manipulation with Selenium. In this tutorial we are going to learn how can we read and write on excel file with the help of Java IO package and Apache POI library.

### **Exporting Excel**

* **How to handle excel file using POI (Maven POM Dependency)**

To read or write an Excel, Apache provide a very famous library POI. This library is capable enough to read and write both **XLS** and **XLSX** file format of excel.

To read **XLS** files an **HSSF** implementation is provided by POI library.

To read **XLSX** , **XSSF** implementation of **POI** **library** will be the choice. Let's study these implementations in detail.

If you are using maven in your project the maven dependency will be

**<dependency>**

**<groupId>org.apache.poi</groupId>**

**<artifactId>poi</artifactId>**

**<version>3.9</version>**

**</dependency>**

Or you can simply download the latest version POI jars from <http://poi.apache.org/download.html> & download [poi-bin-3.10-FINAL-20140208.zip](http://www.apache.org/dyn/closer.cgi/poi/release/bin/poi-bin-3.10-FINAL-20140208.zip)

When you download the zip file for this jar , you need to unzip it and add these all jars to the class path of your project.

### **Classes and Interfaces in POI:**

Following is a list of different Java Interfaces and classes in **POI** for reading **XLS** and **XLSX** file-

* **Workbook -** XSSFWorkbook and HSSFWorkbook classes implement this interface.
* **XSSFWorkbook -** Is a class representation of XLSX file.
* **HSSFWorkbook -** Is a class representation of XLS file.
* **Sheet-** XSSFSheet and HSSFSheet classes implement this interface.
* **XSSFSheet-** Is a class representing a sheet in a XLSX file.
* **HSSFSheet-** Is a class representing a sheet in a XLS file.
* **Row-** XSSFRow and HSSFRow classes implement this interface.
* **XSSFRow-** Is a class representing a row in sheet of XLSX file.
* **HSSFRow-** Is a class representing a row in sheet of XLS file.
* **Cell-** XSSFRow and HSSFRow classes implement this interface.
* **XSSFCell-** Is a class representing a cell in a row of XLSX file.
* **HSSFCell-** Is a class representing a cell in a row of XLS file.

|  |  |
| --- | --- |
|  |  |

Note: We are not using the TestNG [framework](http://www.guru99.com/quick-test-professional-qtp-tutorial-34.html) here. Run the class as Java Application

* **Write data on Excel file**

Complete Example: Here we are trying to write data from excel file by adding new row in excel file