**Links ( Full , Partial)**

**Textbox -- Done**

**Checkbox -- Done**

**Windows --Done**

**Pop up :**

**Windows OS pop up (System popup) – selenium cannot handle**

**Java script popup (Alerts) --Done**

**Radio button -- --Done**

**Dropdown --Done**

**List box(Multi select) --Done**

**Web Tables --Done**

**Frames --Done**

**Advanced controls (Drag and Drop, Slider) ---Done**

**Sikuli – not covered**

**WebElement methods**

**isDisplayed()**

**isEnabled()**

**isSelected() --Done**

**Java script executor**

**Datepicker**

**Textbox :**

**package digitest.app;**

**import java.time.Duration;**

**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.support.ui.ExpectedConditions;**

**import org.openqa.selenium.support.ui.WebDriverWait;**

**public class TextboxDemo {**

**public static void main(String[] args) {**

**// TODO Auto-generated method stub**

**System.setProperty("webdriver.chrome.driver", "C:\\SeleniumDrivers\\chromedriver\\chromedriver.exe");**

**WebDriver driver=new ChromeDriver();**

**String appurl="https://adactinhotelapp.com/";**

**/\***

**driver.get(appurl);**

**//String txt\_username="username";**

**//thread.sleep(2000);**

**// locator used is id**

**driver.findElement(By.id("username")).sendKeys("test");**

**appurl="http://only-testing-blog.blogspot.com/2014/01/textbox.html";**

**driver.get(appurl);**

**// locator used is name**

**driver.findElement(By.name("fname")).sendKeys("selenium");**

**\*/**

**// locator used is classname**

**appurl="https://adactinhotelapp.com/Register.php";**

**driver.get(appurl);**

**//WebElement txt\_username= driver.findElement(By.className("reg\_input"));**

**//Xpath=//tagname[@attribute='value']**

**// WebElement txt\_username= driver.findElement(By.xpath("//input[@id='username']"));**

**//WebElement txt\_username= driver.findElement(By.cssSelector("#username"));**

**WebElement txt\_username= driver.findElement(By.cssSelector(".reg\_input"));**

**WebDriverWait mywaitvar=new WebDriverWait(driver,Duration.ofSeconds(30));**

**mywaitvar.until(ExpectedConditions.visibilityOf(txt\_username));**

**txt\_username.sendKeys("test1");**

**//driver.findElement(By.xpath("//input[@id=username")).sendKeys("test1");**

**driver.close();**

**}**

**}**

**Checkbox :**

**package digitest.app;**

**import org.openqa.selenium.By;**

**import org.openqa.selenium.WebDriver;**

**import org.openqa.selenium.chrome.ChromeDriver;**

**import org.openqa.selenium.edge.EdgeDriver;**

**public class CheckBox {**

**public static void main(String[] args) throws InterruptedException {**

**// TODO Auto-generated method stub**

**//System.setProperty("webdriver.chrome.driver", "C:\\SeleniumDrivers\\chromedriver\\chromedriver.exe");**

**//WebDriver driver=new ChromeDriver();**

**System.setProperty("webdriver.edge.driver","C:\\SeleniumDrivers\\edgedriver\\msedgeadriver.exe");**

**WebDriver driver = new EdgeDriver();**

**String appurl="https://adactinhotelapp.com/";**

**driver.get(appurl);**

**driver.findElement(By.partialLinkText("New User")).click();;**

**System.out.println(driver.getTitle());**

**Thread.sleep(1000);**

**driver.findElement(By.id("tnc\_box")).click();**

**}**

**}**

**Alerts :**

**package digitest.app;**

**import org.openqa.selenium.Alert;**

**import org.openqa.selenium.By;**

**import org.openqa.selenium.WebDriver;**

**import org.openqa.selenium.WebElement;**

**import org.openqa.selenium.chrome.ChromeDriver;**

**import java.util.concurrent.TimeUnit;**

**public class AlertDemo {**

**public static void main(String[] args) throws InterruptedException {**

**// TODO Auto-generated method stub**

**System.setProperty("webdriver.chrome.driver", "C:\\SeleniumDrivers\\chromedriver\\chromedriver.exe");**

**WebDriver driver=new ChromeDriver();**

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

**String appurl="http://register.rediff.com/register/register.php?FormName=user\_details";**

**driver.get(appurl);**

**//input[contains(text(),'name')]**

**WebElement txtname=driver.findElement(By.xpath("//\*[contains(@name,'name')]"));**

**WebElement btnSubmit=driver.findElement(By.id("Register"));**

**btnSubmit.click();**

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

**System.out.println(alert.getText());**

**Thread.sleep(2000);**

**alert.accept();**

**txtname.sendKeys("test");**

**}**

**}**

**Windows :**

**package digitest.app;**

**import java.util.Iterator;**

**import java.util.Set;**

**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;**

**public class HandleWindows {**

**public static void main(String[] args) throws InterruptedException {**

**// TODO Auto-generated method stub**

**System.setProperty("webdriver.chrome.driver", "C:\\SeleniumDrivers\\chromedriver\\chromedriver.exe");**

**WebDriver driver=new ChromeDriver();**

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

**String url="http://only-testing-blog.blogspot.com/2014/01/textbox.html";**

**driver.get(url);**

**WebElement lnk=driver.findElement(By.linkText("Open New Page"));**

**//WebElement txt\_fname=driver.findElement(By.id("text1"));**

**lnk.click();**

**Thread.sleep(2000);**

**Set < String > s = driver.getWindowHandles();**

**Iterator < String > it = s.iterator();**

**it.next(); //parent window**

**String w2nd = it.next(); //child window**

**driver.switchTo().window(w2nd);**

**Thread.sleep(2000);**

**WebElement txt\_fname=driver.findElement(By.id("fname"));**

**txt\_fname.sendKeys("test");**

**driver.close();**

**}**

**}**

**Radio button**

****

**HTML Source**

**<input** type="radio" name="gender" value="male" id="radio\_male" checked="true\

"**>**Male**<br>**

**<input** type="radio" name="gender" value="female" id="radio\_female"**>**Female

<input id="bmwradio" value="bmw" name="cars" type="radio">

@attribute =’value’

//input[@name=’cars’ and @value=’bmw’]

**Select a radio button**

driver.findElement(By.xpath("//input[@name='gender' and @value='female']")).\

click();

Thread.sleep(500);

driver.findElement(By.xpath("//input[@name='gender' and @value='male']")).cl\

ick();

**Clear radio option selection**

Once a radio button is selected, you cannot just clear the selection in Selenium. (Watir,

another test framework, can clear radio selection). You need to select another radio button.

The test script below will throw an error: “invalid element state: Element must be usereditable

in order to clear it.”

driver.findElement(By.xpath("//input[@name='gender' and @value='female']")).\

click();

**try** {

driver.findElement(By.xpath("//input[@name='gender' and @value='female']\

")).clear();

} **catch** (Exception ex) {

*// Selenium does not allow*

System.out.println("Selenium does not allow clear currently selected rad\

io button, just select another one");

driver.findElement(By.xpath("//input[@name='gender' and @value='male']")\

).click();

}

**Assert a radio option is selected**

driver.findElement(By.xpath("//input[@name='gender' and @value='female']")).\

click();

**assert** driver.findElement(By.xpath("//input[@name='gender' and @value='femal\

e']")).isSelected();

**assert** !driver.findElement(By.xpath("//input[@name='gender' and @value='male\

']")).isSelected();

**Iterate radio buttons in a radio group**

So far we have been focusing on identifying web controls by using one type of locator find-

Element. Here I introduce another type of locator (I call them plural locators): findElements.

**assert** driver.findElements(By.name("gender")).size() == 2;

**for** (WebElement rb : driver.findElements(By.name("gender"))) {

**if** (rb.getAttribute("value").equals("female")) {

rb.click();

}

}

Example :

**package** digitest.app;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.PageLoadStrategy;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.chrome.ChromeOptions;

**public** **class** RadiobuttonDemo {

**public** **static** **void** main(String[] args) **throws** InterruptedException {

// **TODO** Auto-generated method stub

ChromeOptions chromeOptions;

WebDriver driver;

String appurl;

// **TODO** Auto-generated method stub

System.***out***.println("Launch browser");

System.*setProperty*("webdriver.chrome.driver", "C:\\SeleniumDrivers\\chromedriver\\chromedriver.exe");

chromeOptions = **new** ChromeOptions();

chromeOptions.setPageLoadStrategy(PageLoadStrategy.***EAGER***);

chromeOptions.addArguments("--remote-allow-origins=\*");

driver = **new** ChromeDriver(chromeOptions);

appurl = "https://www.letskodeit.com/practice";

driver.manage().timeouts().~~implicitlyWait~~(10, TimeUnit.***SECONDS***);

driver.get(appurl);

/\*

WebElement rdbutton1=driver.findElement(By.xpath("//input[@name='cars' and @value='honda']"));

rdbutton1.click();

System.out.println(rdbutton1.isSelected());

WebElement rdbutton2=driver.findElement(By.xpath("//input[@name='cars' and @value='bmw']"));

System.out.println(rdbutton2.isSelected());

\*/

//multiple radio button

**for** (WebElement rb : driver.findElements(By.*xpath*("//input[@name='cars' and @type='radio']"))) {

rb.click();

System.***out***.println(rb.isSelected());

Thread.*sleep*(2000);

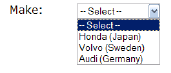
}

}

}

**Select List**

A Select list is also known as a drop-down list or combobox.



**HTML Source**

**<select** name="car\_make" id="car\_make\_select"**>**

**<option** value=""**>**-- Select --**</option>**

**<option** value="honda"**>**Honda (Japan)**</option>**

**<option** value="volvo"**>**Volvo (Sweden)**</option>**

**<option** value="audi"**>**Audi (Germany)**</option>**

**</select>**

**Select an option by text**

The label of a select list is what we can see in the browser.

Select select = **new** Select(driver.findElement(By.name("car\_make")));

select.selectByVisibleText("Volvo (Sweden)");

**Select an option by value**

The value of a select list is what to be passed to the server.

Select select = **new** Select(driver.findElement(By.id("car\_make\_select")));

select.selectByValue("audi");

**Select an option by iterating all options**

Here I will show you a far more complex way to select an option in a select list, not for the

sake of complexity, of course. A select list contains options, where each option itself is a valid

control in Selenium.

WebElement selectElem = driver.findElement(By.id("car\_make\_select"));

**for** (WebElement option : selectElem.findElements(By.tagName("option"))) {

**if** (option.getText().equals("Volvo (Sweden)")) {

option.click();

}

}

Example :

**package** digitest.app;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.PageLoadStrategy;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.chrome.ChromeOptions;

**import** org.openqa.selenium.support.ui.Select;

**public** **class** DropdownDemo {

**public** **static** **void** main(String[] args) **throws** InterruptedException {

// **TODO** Auto-generated method stub

ChromeOptions chromeOptions;

WebDriver driver;

String appurl;

// **TODO** Auto-generated method stub

System.***out***.println("Launch browser");

System.*setProperty*("webdriver.chrome.driver", "C:\\SeleniumDrivers\\chromedriver\\chromedriver.exe");

chromeOptions = **new** ChromeOptions();

chromeOptions.setPageLoadStrategy(PageLoadStrategy.***EAGER***);

chromeOptions.addArguments("--remote-allow-origins=\*");

driver = **new** ChromeDriver(chromeOptions);

appurl = "https://www.letskodeit.com/practice";

driver.manage().timeouts().~~implicitlyWait~~(10, TimeUnit.***SECONDS***);

driver.get(appurl);

/\*

// code to select individual item

WebElement ddCars=driver.findElement(By.id("carselect"));

Select s1=new Select(ddCars);

s1.selectByVisibleText("Honda");

Thread.sleep(2000);

s1.selectByIndex(0);

Thread.sleep(2000);

s1.selectByValue("benz");

\*/

//code to select all items in dropdown

WebElement selectElem = driver.findElement(By.*id*("carselect"));

**for** (WebElement option : selectElem.findElements(By.*tagName*("option"))) {

// if (option.getText().equals("Volvo (Sweden)")) {

option.click();

Thread.*sleep*(2000);

// }

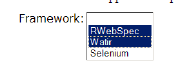
}

}

}

**Select multiple options**

A select list also supports multiple selections.



**HTML Source**

**<select** id="framework\_select" name="test\_framework" multiple="multiple"**>**

**<option></option>**

**<option** value="rwebspec"**>**RWebSpec**</option>**

**<option** value="watir"**>**Watir**</option>**

**<option** value="selenium"**>**Selenium**</option>**

**</select>**

**Test Script**

Select select = **new** Select(driver.findElement(By.name("test\_framework")));

select.selectByVisibleText("Selenium");

select.selectByValue("rwebspec");

select.selectByIndex(2);

**assert** select.getAllSelectedOptions().size() == 3;

**Clear one selection**

Select select = **new** Select(driver.findElement(By.name("test\_framework")));

select.selectByVisibleText("RWebSpec");

select.selectByVisibleText("Selenium");

select.deselectByVisibleText("RWebSpec");

select.deselectByValue("selenium");

*// one more*

*// select.deselectByIndex(0);*

**assert** select.getAllSelectedOptions().size() == 0;

**Clear selection**

Clear selection works the same way for both single and multiple select lists.

Select select = **new** Select(driver.findElement(By.name("test\_framework")));

select.selectByVisibleText("Selenium");

select.selectByVisibleText("RWebSpec");

select.deselectAll();

**assert** select.getAllSelectedOptions().size() == 0;

**Assert selected option**

To verify a particular option is currently selected in a select list:

Select select = **new** Select(driver.findElement(By.id("car\_make\_select")));

select.selectByValue("audi");

**assert** "Audi (Germany)".equals(select.getFirstSelectedOption().getText());

**public** **class** MultiSelect {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

ChromeOptions chromeOptions;

WebDriver driver;

String appurl;

// **TODO** Auto-generated method stub

System.***out***.println("Launch browser");

System.*setProperty*("webdriver.chrome.driver", "C:\\SeleniumDrivers\\chromedriver\\chromedriver.exe");

chromeOptions = **new** ChromeOptions();

chromeOptions.setPageLoadStrategy(PageLoadStrategy.***EAGER***);

chromeOptions.addArguments("--remote-allow-origins=\*");

driver = **new** ChromeDriver(chromeOptions);

appurl = "https://www.letskodeit.com/practice";

driver.manage().timeouts().~~implicitlyWait~~(10, TimeUnit.***SECONDS***);

driver.get(appurl);

// code to select individual item

WebElement ddfruits=driver.findElement(By.*id*("multiple-select-example"));

Select s1=**new** Select(ddfruits);

s1.selectByIndex(1);

s1.selectByIndex(2);

System.***out***.println(s1.getAllSelectedOptions().size());

s1.deselectByIndex(2);

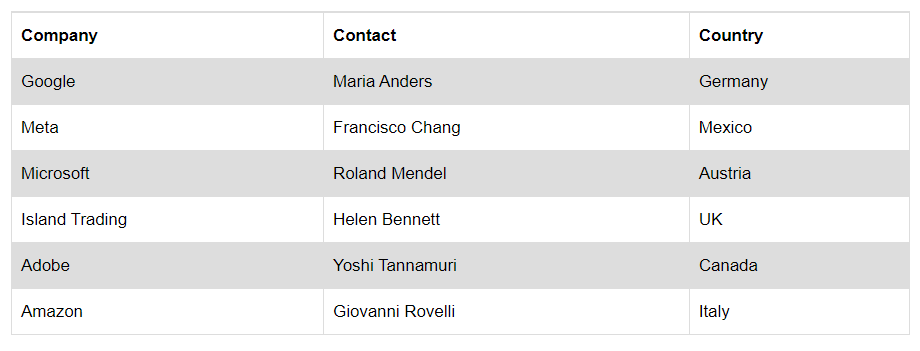
System.***out***.println(s1.getAllSelectedOptions().size());

System.***out***.println(s1.getFirstSelectedOption().getText());

}

}

**Handle Static and Dynamic Web Table in Selenium WebDriver**



**1. What is Web Table?**

* Types of Web Table
  + Static Table
  + Dynamic Table

**2. Automate Reading data from Static Web Table with Selenium**

* Practice Exercises for automating Static Table
* Solution Code for Static Table
* Code Explanation

**3. Automate Handling Dynamic Web Table with Selenium**

* Practice Exercises for automating Dynamic Table
* Solution Code for Dynamic Table
* Code Explanation

**1. What is a Web Table?**

A web table is a way of representing data in rows and columns. It's the html representation of MS-Excel table. Web table has various html tags that automation engineer should be aware of like., table, th, tr, td. Let's understand these tags a bit more:

* "<**table**>" - It defines a table. You can also say that it's the starting point of a table.
* "<**th**>" - It defines a header cell, which means you should define your headings inside th tag.
* "<**tr**>" - It defines a row in a table.
* "<**td**>" - It defines a cell in a table. "td" always lie inside the tr tag.

#### 1.1. Types of Web Table

There are two types of web table., Static and Dynamic. Let's read more about both.

#### 1.1.1. Static Table

As the name suggests, static table is a table which displays the static data. It means the data that it will display will not change on reloading the web page. So, in that case we know that there will r no. of rows and c no. of columns inside it.

#### 1.1.2. Dynamic Table

First of all let's understand that what we call a dynamic table. So here is the answer, there might occur some scenarios where every time a page loads with different number of rows and columns or different columns for each row. When we don't know that how the expected data outcome will be presented in a table, that's called a dynamic table.

**public** **class** **Table** {

WebDriver driver = **new** FirefoxDriver();

**@BeforeTest**

**public** **void** **setup**() **throws** Exception {

    driver.manage().window().maximize();

    driver.manage().timeouts().implicitlyWait(**15**, TimeUnit.SECONDS);

  driver.get("<https://www.techlistic.com/p/demo-selenium-practice.html>");

}

**@AfterTest**

**public** **void** **tearDown**() **throws** Exception {

     driver.quit();

}

**@Test**

**public** **void** **print\_data**(){

//Get number of rows In table.

**int** Row\_count = driver.findElements(By.xpath("//\*[@id='post']/div[1]/table/tbody/tr")).size();

System.out.println("Number Of Rows = "+Row\_count);

//Get number of columns In table.

**int** Col\_count = driver.findElements(By.xpath("//\*[@id='post']/div[1]/table/tbody/tr[1]/td")).size();

System.out.println("Number Of Columns = "+Col\_count);

//divided xpath In three parts to pass Row\_count and Col\_count values.

String first\_part = "//\*[@id='post']/div[1]/table/tbody/tr[";

String second\_part = "]/td[";

String third\_part = "]";

//Used for loop for number of rows.

**for** (**int** i=**1**; i&lt;=Row\_count; i++){

//Used for loop for number of columns.

**for**(**int** j=**1**; j&lt;=Col\_count; j++){

//Prepared final xpath of specific cell as per values of i and j.

String final\_xpath = first\_part+i+second\_part+j+third\_part;

//Will retrieve value from located cell and print It.

String Table\_data = driver.findElement(By.xpath(final\_xpath)).getText();

System.out.print(Table\_data +" ");

}

System.out.println("");

System.out.println("");

}

}

}

#### Code Explanation step-by-step:

1. In setup method do all the config stuff, like launch a browser, open url, set wait etc.
2. In test method, findElements() method is used with xpath of the first table row (tr). It will extract all the table row elements and return a list.
3. You can see, we have used size() method at the end of first line of test method. This would return size of the list. So, it's actually returning the no. of rows present on table.
4. Similarly, we get table columns using the findElements() method along with first table column xpath (tc). And then use size() method to get the no. of columns.
5. Store the no. of rows and columns in integer variables.
6. Now we create custom xpath for table row and column, we replace row and column index to variables. So, that we can iterate over every table row's td (cell) using for loop.
7. We have used first for loop row count, which will iterate over every row.
8. And second for loop will iterate over every table data (td) of each row one by one.
9. So, loops works in following fashion:
   1. First loop will be initialized with row, means i=1.
   2. Now, execution will enter in second loop, and this loop will iterate over every column element (td) present in the first row. So, here i will remain 1 but value of j keeps on changing by +1. After the value of j reached the total no. of columns present in table, second loop exits.
   3. And now, the execution again comes to first loop, and the value of i will be incremented by 1 and i becomes 2 second time.
   4. Step will be executed again and so on until the value of i reaches the total no. of rows present in table.
10. **public** **class** WebTableDemo {
11. **public** **static** **void** main(String[] args) {
12. // **TODO** Auto-generated method stub
13. // **TODO** Auto-generated method stub
14. ChromeOptions chromeOptions;
15. WebDriver driver;
16. String appurl;
17. // **TODO** Auto-generated method stub
18. System.***out***.println("Launch browser");
19. System.*setProperty*("webdriver.chrome.driver", "C:\\SeleniumDrivers\\chromedriver\\chromedriver.exe");
20. chromeOptions = **new** ChromeOptions();
21. chromeOptions.setPageLoadStrategy(PageLoadStrategy.***EAGER***);
22. chromeOptions.addArguments("--remote-allow-origins=\*");
23. driver = **new** ChromeDriver(chromeOptions);
24. appurl = "https://www.letskodeit.com/practice";
25. driver.manage().timeouts().~~implicitlyWait~~(10, TimeUnit.***SECONDS***);
26. driver.get(appurl);
27. /\*
28. rows and columns
30. htmls --- table , tr ,td , th
32. th = header
33. tr = row
34. td =column , cell
35. \*/
37. //Step 1 : Identify rows and columns of a table
38. // no of rows in a table
39. **int** row\_count =driver.findElements(By.*xpath*("//table[@id='product']/tbody/tr")).size();
40. System.***out***.println("Number of rows :"+row\_count);
41. // no of columns in a table
42. **int** col\_count=driver.findElements(By.*xpath*("//table[@id='product']/tbody/tr[2]/td")).size();
43. System.***out***.println("Number of columns :"+col\_count);
45. String first\_part="//table[@id='product']/tbody/tr[";
46. String second\_part="]/td[";
47. String third\_part="]";
48. **for** (**int** i=2;i<=row\_count;i++)
49. {
50. **for** (**int** j=1;j<=col\_count;j++)
51. {
52. System.***out***.print(driver.findElement(By.*xpath*(first\_part+i+second\_part+j+third\_part)).getText());
53. System.***out***.print("\u0009");
54. }
55. System.***out***.println("");
57. }
58. }
59. }

**3. Automate Dynamic Web Table with Selenium**

Selenium doesn't provide us any specific command for handling a dynamic table. This is a special case in automation, so to handle this case we have to write the logic inside our code to handle the dynamic loading of data. It will require both Selenium and Coding skills.

For that, we have to put two for loops,

1. First loop will iterate all the rows.  
2. Second loop inside the first loop and it will fetch the no. of columns present in that particular row and then iterate the columns.

**public** **class** **DynamicTable** {

WebDriver driver = **new** FirefoxDriver();

**@BeforeTest**

**public** **void** **setup**() **throws** Exception {

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(**5**, TimeUnit.SECONDS);

driver.get("<https://www.techlistic.com/p/demo-selenium-practice.html>");

}

**@AfterTest**

**public** **void** **tearDown**() **throws** Exception {

    driver.quit();

}

**@Test**

**public** **void** **test\_Dynamic\_Webtable**() {

// Locate table

WebElement mytable = driver.findElement(By.xpath(".//\*[@id='post']/div[1]/table/tbody"));

// Locate rows of table and save locators of each row in a list

List rows\_table = mytable.findElements(By.tagName("tr"));

// Get no. of rows in table

**int** rows\_count = rows\_table.size();

// Loop will execute till the last row of table

**for** (**int** row=**0**; row&lt;=rows\_count; row++)

{

// locate columns(cells) of that specific row.

List Columns\_row = rows\_table.get(row).findElements(By.tagName("td"));

// Get no. of columns(cells) In that specific row.

**int** columns\_count = Columns\_row.size();

System.out.println("Number of cells In Row "+row+" are "+columns\_count);

// Loop will execute till the last cell of that specific row.

**for** (**int** column=**0**; column&lt;=columns\_count; column++) {

// Retrieve text from that specific cell.

String celtext = Columns\_row.get(column).getText();

System.out.println("Cell Value Of row number "+row+" and column number "+column+" Is "+celtext);

}

}

}

}

#### Code Explanation step-by-step:

1. All steps are exactly same wrt the 2.2 section of this post, except one step.
2. In case of a static table we are fetching the no. of columns before starting any for loop. But in case of dynamic table, we are fetching columns after first for loop has started.
3. It means that we are fetching the columns for every row separately because we don't know while writing the code that each row would have how many columns.
4. So, this is the way we are handling the dynamic web table in Selenium.

**Frames**

HTML Frames are treated as independent pages, which is not a good web design practice. As

a result, few new sites use frames nowadays. However, there a quite a number of sites that

uses iframes.

**Testing Frames**

Here is a layout of a fairly common frame setup: navigations on the top, menus on the left

and the main content on the right.



**HTML Source**

**<frameset** rows="100,\*" frameborder="0" border="0" framespacing="0"**>**

**<frame** name="topNav" src="top\_nav.html"**>**

**<frameset** cols="200,\*" frameborder="0" border="0" framespacing="0"**>**

**<frame** name="menu" id="menu\_frame" src="menu\_1.html" marginheight="0" ma\

rginwidth="0" scrolling="auto" noresize**>**

**<frame** name="content" src="content.html" marginheight="0" marginwidth="0\

" scrolling="auto" noresize**>**

**</frameset>**

**</frameset>**

To test a frame with Selenium, we need to identify the frame first by ID or NAME, and then

switch the focus on it. The test steps after will be executed in the context of selected frame.

Use switch\_to.default\_content() to get back to the page (which contains frames).

driver.switchTo().frame("topNav"); *// name*

driver.findElement(By.linkText("Menu 2 in top frame")).click();

*// need to switch to default before another switch*

driver.switchTo().defaultContent();

driver.switchTo().frame("menu\_frame"); *//fail on Chrome, fine for Firefox*

driver.findElement(By.linkText("Green Page")).click();

driver.switchTo().defaultContent();

driver.switchTo().frame("content");

driver.findElement(By.linkText("Back to original page")).click();

This script clicks a link in each of three frames: top, left menu and content.

**Testing iframe**

An iframe (Inline Frame) is an HTML document embedded inside another HTML document

on a web site.

**Example page**



**HTML Source**

**<IFRAME** frameborder='1' id="Frame1" src="login\_iframe.html"

Style="HEIGHT: 100px; WIDTH: 320px; MARGIN=0" SCROLLING="no" **>**

**</IFRAME>**

The test script below enters text in the main page, fills the sign in form in an iframe, and

ticks the checkbox on the main page:

Frames 60

driver.switchTo().frame("Frame1"); *// name*

driver.findElement(By.name("username")).sendKeys("tester");

driver.findElement(By.name("password")).sendKeys("TestWise");

driver.findElement(By.id("loginBtn")).click();

**assert** driver.getPageSource().contains("Signed in");

driver.switchTo().defaultContent();

driver.findElement(By.id("accept\_terms")).click();

**package** digitest.app;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.PageLoadStrategy;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.chrome.ChromeOptions;

**public** **class** FrameDemo {

**public** **static** **void** main(String[] args) **throws** InterruptedException {

// **TODO** Auto-generated method stub

ChromeOptions chromeOptions;

WebDriver driver;

String appurl;

// **TODO** Auto-generated method stub

System.***out***.println("Launch browser");

System.*setProperty*("webdriver.chrome.driver", "C:\\SeleniumDrivers\\chromedriver\\chromedriver.exe");

chromeOptions = **new** ChromeOptions();

chromeOptions.setPageLoadStrategy(PageLoadStrategy.***EAGER***);

chromeOptions.addArguments("--remote-allow-origins=\*");

driver = **new** ChromeDriver(chromeOptions);

appurl = "https://www.letskodeit.com/practice";

driver.manage().timeouts().~~implicitlyWait~~(10, TimeUnit.***SECONDS***);

driver.get(appurl);

driver.switchTo().frame("courses-iframe");

WebElement txtSearch=driver.findElement(By.*xpath*("//a[contains(text(),'Sign In')]"));

txtSearch.click();

System.***out***.println("success");

driver.switchTo().defaultContent();

**for** (WebElement rb : driver.findElements(By.*xpath*("//input[@name='cars' and @type='radio']"))) {

rb.click();

System.***out***.println(rb.isSelected());

Thread.*sleep*(2000);

}

}

}

**Advanced User Interactions**

The Actions in Selenium WebDriver provides a way to set up and perform complex user

interactions. Specifically, grouping a series of keyboard and mouse operations and sending

to the browser.

**Mouse interactions**

• click()

• clickAndHold()

• contextClick()

• doubleClick()

• dragAndDrop()

• dragAndDropBy()

• moveByOffset()

• moveToElement()

• release()

**Keyboard interactions**

• keyDown()

• keyUp()

• sendKeys()

**The usage**

new Actions(driver). + one or more above operations + .perform();

Import the classes below in your test scripts to use Actions.

**import org.openqa.selenium.interactions.Action**;

**import org.openqa.selenium.interactions.Actions**;

**Double click a control**

WebElement elem = driver.findElement(By.id("pass"));

Actions builder = **new** Actions(driver);

builder.doubleClick(elem).perform();

**Move mouse to a control - Mouse Over**

WebElement elem = driver.findElement(By.id("email"));

Actions builder = **new** Actions(driver);

builder.moveToElement(elem).perform();

**Click and hold - select multiple items**

The test scripts below clicks and hold to select three controls in a grid.

driver.get("http://jqueryui.com/selectable");

driver.findElement(By.linkText("Display as grid")).click();

Thread.sleep(500);

driver.switchTo().frame(0);

List<WebElement> listItems = driver.findElements(By.xpath("//ol[@id='selecta\

ble']/li"));

Actions builder = **new** Actions(driver);

builder.clickAndHold(listItems.get(1))

.clickAndHold(listItems.get(3))

.click()

.perform();

driver.switchTo().defaultContent();

<https://jqueryui.com/draggable/>

**Drag and drop**

Drag-n-drop is increasingly more common in new web sites. Testing this feature can be

largely achieved in Selenium, I used the word ‘largely’ means achieving the same outcome,

but not the ‘mouse dragging’ part. For this example page,



the test script below will *drop* ‘Item 1’ to ‘Trash’.

driver.get(TestHelper.siteUrl() + "drag\_n\_drop.html");

WebElement dragFrom = driver.findElement(By.id("item\_1"));

WebElement target = driver.findElement(By.id("trash"));

Actions builder = **new** Actions(driver);

Action dragAndDrop = builder.clickAndHold(dragFrom)

.moveToElement(target)

.release(target)

.build();

dragAndDrop.perform();

Example 1:

driver.get("https://jqueryui.com/resources/demos/droppable/default.html");

driver.manage().window().maximize();

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

WebElement draggable = driver.findElement(By.id("draggable"));

WebElement droppable = driver.findElement(By.id("droppable"));

Actions action = new Actions(driver);

action.dragAndDrop(draggable, droppable).perform();

Assignment :

<https://jqueryui.com/droppable/>

**Drag slider**

Slider (a part of JQuery UI library) provide users an very intuitive way to adjust values

(typically in settings).



WebElement mainSlider = driver.findElement(By.*id*("slider"));

**int** width = mainSlider.getSize().width/2;

WebElement slider = driver.findElement(By.*xpath*("//\*[@id=\"slider\"]/span"));

**new** Actions(driver).dragAndDropBy(slider, width, 0).perform();

//\* Y coordinate is zero

Example01 :

driver = **new** ChromeDriver(chromeOptions);

appurl = "https://jqueryui.com/slider/";

driver.manage().timeouts().~~implicitlyWait~~(10, TimeUnit.***SECONDS***);

driver.get(appurl);

driver.switchTo().frame(0);

WebElement elem = driver.findElement(By.*id*("slider"));

Actions move = **new** Actions(driver);

Action action = (Action) move.dragAndDropBy(elem, 10, 0).build();

action.perform();

Resizable :

WebElement resizable = driver.findElement(By.xpath("//\*[@id=\"resizable\"]/div[3]"));

Actions action = new Actions(driver);

action.dragAndDropBy(resizable, 400, 400).perform();

**Keyboard interactions**

ChromeOptions chromeOptions;

WebDriver driver;

String appurl;

// **TODO** Auto-generated method stub

System.***out***.println("Launch browser");

System.*setProperty*("webdriver.chrome.driver", "C:\\SeleniumDrivers\\chromedriver\\chromedriver.exe");

chromeOptions = **new** ChromeOptions();

chromeOptions.setPageLoadStrategy(PageLoadStrategy.***EAGER***);

chromeOptions.addArguments("--remote-allow-origins=\*");

driver = **new** ChromeDriver(chromeOptions);

appurl = "https://demoqa.com/text-box";

driver.manage().timeouts().~~implicitlyWait~~(10, TimeUnit.***SECONDS***);

driver.get(appurl);

driver.manage().window().maximize();

WebElement currentAddress=driver.findElement(By.*id*("currentAddress"));

currentAddress.sendKeys("43 School Lane London EC71 9GO");

currentAddress.click();

// Copy the Current Address

currentAddress.sendKeys(Keys.***CONTROL***);

currentAddress.sendKeys("A");

currentAddress.sendKeys(Keys.***CONTROL***);

currentAddress.sendKeys("C");

//Press the TAB Key to Switch Focus to Permanent Address

currentAddress.sendKeys(Keys.***TAB***);

Thread.*sleep*(2000);

//Paste the Address in the Permanent Address field

WebElement permanentAddress=driver.findElement(By.*id*("permanentAddress"));

permanentAddress.sendKeys(Keys.***CONTROL***);

permanentAddress.sendKeys("V");

Executing JavaScript using the JavascriptExecutor interface

driver = new ChromeDriver();

driver.get("https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\_submit\_get");

driver.manage().window().maximize();

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

/\*Set<String> winids = driver.getWindowHandles();

Iterator<String> iterate = winids.iterator();

String first\_window = iterate.next();

\*/

driver.switchTo().frame("iframeResult");

//driver.findElement(By.xpath("/html/body/button")).click();

//((JavascriptExecutor) driver).executeScript("myFunction()");

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("myFunction()");

js.executeScript("arguments[0].style.border='3px solid red'", driver.findElement(By.id("mySubmit")));