**Handling Web Tables, Frames, and Dynamic Elements in Selenium Script**

we would discuss about***handling Web tables, iframe and dynamic elements***which are essential part of any web project.

This module consists of 3 different topics and their handling mechanisms in selenium script.

1. **Web Tables/HTML tables**
2. **Frames**
3. **Dynamic elements**

### ****#1) Web Tables/HTML Tables****

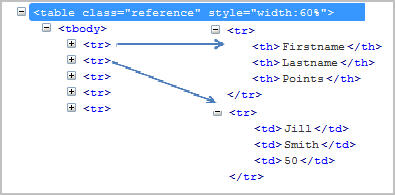
In this module we will learn about the web tables or html tables in a web page, tags available in html and how to handle web tables dynamically.

Web tables are basically group of elements that are logically stored in a row and column format. It is used to organize similar information in a web page.

**Below is an example of Html table:**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Handling-web-tables-in-selenium.jpg)

**Below is the snippet of html structure of an html table:**

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Handling-web-tables-in-selenium-1.jpg)

**Below tags are generally defined in an html tables:**

1.’table’ tag defines html table.  
2.’tbody’ tag defines container for rows and columns.  
3.’tr’ defines rows in an html table.  
4.’td’/’th’ define column of an html table.

**Find the details of a web table:**

There are many ways we can handle a web table.

**Approach #1:**

Below is the xpath of one of the cell in html table. Let’s say “firstname”

//div[@id=’main’]/table[1]/tbody/tr[1]/th[1]

tr[1] defines first row and th[1] defines first column.

If number of rows and columns are always constant, let’s say our html table will always have 5 rows and 3 columns.

|  |  |  |
| --- | --- | --- |
| 1 | for(int numberOfRows=1; numberOfRows<=5; numberOfRows++) | |
| 2 | { |

|  |  |  |
| --- | --- | --- |
| 3 | for(int numberOfCol=1; numberOfCol <=3; numberOfCol++) | |
| 4 | { |

|  |  |  |
| --- | --- | --- |
| 5 | System.out.println(driver.findElement(By.xpath | |
| 6 | (“//div[@id='main']/table[1]/tbody/tr |

|  |  |  |
| --- | --- | --- |
| 7 | [“+numberOfRows+”]/th[“+numberOfCol+”]”))); | |
| 8 | } |

|  |  |
| --- | --- |
| 9 | } |

Except row and column number, each component of xpath remains the same. So you can iterate using “for loop” for each row and column as mentioned above.

**Approach #2**:

First approach is best suitable for the table which doesn’t change its dimensions and always remains the same. Above approach will not be a perfect solution for dynamically changing web tables.

**Let’s take above html table as an example:**

|  |  |  |
| --- | --- | --- |
| 1 | WebElement htmltable=driver.findElement(By.xpath("//\*[@id='main']/table[1]/tbody")); | |
| 2 | List<WebElement> rows=htmltable.findElements(By.tagName("tr")); |

|  |  |
| --- | --- |
| 3 |  |
| 4 | for(int rnum=0;rnum<rows.size();rnum++) | |

|  |  |
| --- | --- |
| 5 | { |
| 6 | List<WebElement> columns=rows.get(rnum).findElements(By.tagName("th")); | |

|  |  |  |
| --- | --- | --- |
| 7 | System.out.println("Number of columns:"+columns.size()); | |
| 8 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | for(int cnum=0;cnum<columns.size();cnum++) | | |
| 10 | | { |

|  |  |  |
| --- | --- | --- |
| 11 | System.out.println(columns.get(cnum).getText()); | |
| 12 | } |

|  |  |
| --- | --- |
| 13 | } |

**Step 1**: First get the entire html table and store this in a variable ‘htmltable’ of type web element.

**Step 2**: Get all the rows with tag name ‘tr’ and store all the elements in a list of web elements. Now all the elements with tag ‘tr’ are stored in ‘rows’ list.

**Step 3**: Loop through each row and get the list of elements with tag‘th’. ‘rows.get(0)’ will give first row and‘findElements(By.tagName(“th”))’ will give list of columns for the row.

**Step 4**: Iterate using ‘columns.getsize()’ and get the details of each cell.

**Note**: Above approach will be best suitable if the table dimensions changes dynamically.

This concludes the topic how to handle web tables in selenium. Next we will learn about handling an element inside frame.

### ****#2) Frames:****

In this section we will learn about the frames in a web page and how to identify the frames. Also we will find out how we can handle a frame in selenium WebDriver.

Many developers like to place elements inside frame. Frame is just like a container where few elements can be grouped.

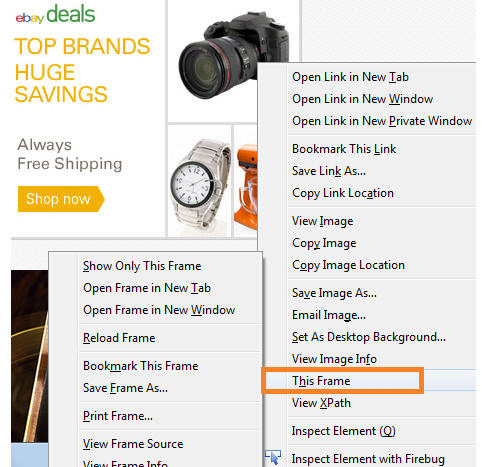
**Identification of a frame:**

Different ways to know if the element is present inside a frame or not

**#1**. Right click on the element. Check if “This Frame” option is available. If This frame option is available, it means that the element is inside a frame.

------------

**#2**. View page source of the web page and check if any tag is available for ‘iframe’.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Handling-iframes-in-selenium.jpg)

**Verify Number of frames in a webpage**:

All the frames are having tag name as “iframe”.

List<WebElement> frameList=driver.findElements(By.tagName(“iframe”));  
System.out.println(frameList.size());

**In above example**: frameList will have all the list of frames andframeList.size() will give the number of frames.

**Handling an element inside frame:**

If an element is inside a frame then control has to switch to frame first and then start operating on the elements.

**Step 1**: To switch inside a frame:

driver.switchTo().frame(1); //pass frame number as parameter.  
or  
driver.switchTo().frame(“frame Name”); //pass frame name as parameter.  
or  
driver.switchTo().frame(“xpath of the frame”);

**Step 2**: After switching inside a frame selenium will be able to operate on elements.

driver.findElement(//\*[@id=’username’]).sendKeys(“username”);  
driver.findElement(//\*[@id=’pass’]).sendKeys(“password”);

Here, we have learned how to handle an element inside frame and next we will cover about the different ways to handle dynamic element.

### ****#3) Dynamic elements:****

In this section we will learn different ways to handle dynamic element and construct generic Xpath.

In few scenarios, element attributes change dynamically. It can be ‘id’, ’name’ etc.

**Example**: let’s say ‘id’ of a username field is ‘username\_123’ and the xpath will be

//\*[@id=’username\_123′] but when you open the page again the ‘id’ of ‘username’ field might have changed and the new value may be ‘username\_234’.

In this case the test will fail because the selenium could not find the xpath you have passed earlier as the id of the field has changed to some other value.

There are many approaches depending upon the type of problem:

**Problem Type 1:**If part of the attribute value changes**.**

**Example**: As in the above example, id value changes but few fields remains constant.  
‘username\_123’ changed to ‘username\_234’ but ‘username’ always remained constant.

**You can construct xpath as below:**

driver.findElement(By.xpath(“//\*[contains(@id,’username’)]”)).sendKeys(“username”);  
driver***.***findElement(By.xpath(“//\*[starts-with(@id,’user’)]”)).sendKeys(“username”);

‘contains’ is a java method which checks if id contains the substring username.  
starts-with() checks if any attribute starts with “user”.

**Problem Type 2:**If entire value of the attribute changes dynamically.

Again in this case, there could be different approaches:

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Handling-Dynamic-elements-in-selenium.jpg)  
**For example**: if id of ‘login’ field changes dynamically and there is no constant value to use contains method.

**Solution**: Use of sendKeys.  
Selenium provides different api to use function keys. For example tab key, enter keys, F5 etc.

**Step 1**: Enter password  
driver.findElement(By.id(“password”)).sendKeys(“password”));

**Step 2**: Use key functions to navigate to element.  
driver.findElement(By.id(“password”)).sendKeys(Keys.ENTER));  
or  
driver.findElement(By.id(“password”)).sendKeys(Keys.TAB));

### ****Conclusion:****

Web tables, frames and dynamic elements are essential part of any web project. It is always desirable to write effective code to handle web tables and dynamic elements.

Understanding the construction of generic xpath which is very helpful while handling dynamic elements. In case of a frame, your script has to switch the frame and then operate on the element.