**1) How to handle ajax calls using selenium web driver.**

AJAX allows the Web page to retrieve small amounts of data from the server without reloading the entire page.

To test Ajax application, different wait methods should be applied

ThreadSleep

Implicit Wait

Explicit Wait

WebdriverWait

Fluent Wait

Creating automated test request may be difficult for testing tools as such AJAX application often use different encoding or serialization technique to submit POST data.

**2) Super most class for error and exception?**

Throwable

**3) How to validate the colors in selenium?**

**Steps:**

First of all, we have to get the value of colour using **getCssValue** method provided by **Selenium Webdriver**.

We can do it by using the below code:

***String color = driver.findElement(By.xpath(“//div[contains(@class, ‘logo-subtext’)]”)).getCssValue(“color”);***

In the above code, CSS attribute **‘color’** is stored in a String variable **‘color’**.

The above code will return value in **RGB** format such as **“rgba(36, 93, 193, 1)”**.

Now we will convert it into **hexadecimal code** using **Java** as shown below:

***String[] hexValue = color.replace(“rgba(“, “”).replace(“)”, “”).split(“,”);***

***hexValue[0] = hexValue[0].trim();***

***int hexValue1 = Integer.parseInt(hexValue[0]);***

***hexValue[1] = hexValue[1].trim();***

***int hexValue2 = Integer.parseInt(hexValue[1]);***

***hexValue[2] = hexValue[2].trim();***

***int hexValue3 = Integer.parseInt(hexValue[2]);***

***String actualColor = String.format(“#%02x%02x%02x”, hexValue1, hexValue2, hexValue3);***

**It is done!!**

You will get the value of colour in Hexadecimal code after executing the above code.

We can add an Assert statement to verify that the colour is matching with the expected colour.

***Assert.assertTrue(actualColor.equals(“#0045d0”));***

***//Expected value is taken from the above screen shot***

After performing all the above steps, your whole method looks like below:

**Usage:**

* You can integrate this colour-verification code in your existing test cases. So there is no need to maintain the separate test cases for colour verification of Elements
* You can verify the colour changing scenarios while hovering the mouse on Elements
* You can verify the colours of the messages **(like error message, warning message, some information, etc.)**

4)How to handle dynamic webelements in selenium.

**#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.

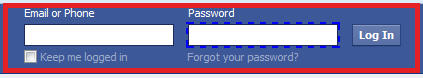
**You can construct xpath as below:**

*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:

  
**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.

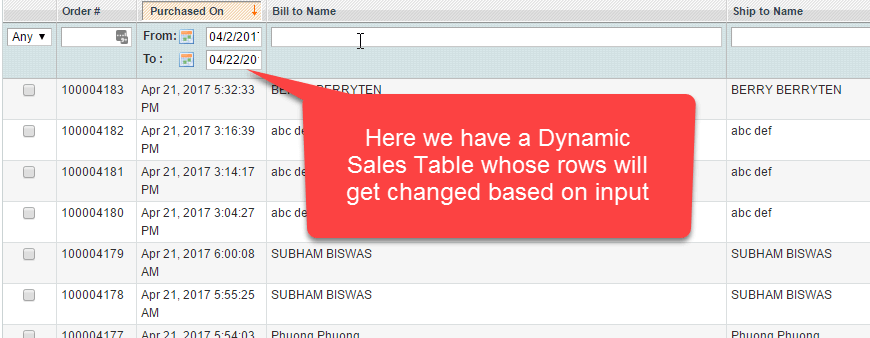
5)

**Handling Dynamic Web Tables Using Selenium WebDriver**

There are two types of HTML tables published on the web-

1. **Static tables**: Data is static i.e. Number of rows and columns are fixed.
2. **Dynamic tables**: Data is dynamic i.e. Number of rows and columns are NOT fixed.

Below is an example of a dynamic table of Sales. Based on input date filters, number of rows will get altered. So, it is dynamic in nature.



Handling static table is easy, but dynamic table is a little bit difficult as rows and columns are not constant.

**In this tutorial, you will learn-**

* [Using X-Path to Locate Web Table Elements](https://www.guru99.com/handling-dynamic-selenium-webdriver.html#1)
* [Example: Fetch number of rows and columns from Dynamic WebTable](https://www.guru99.com/handling-dynamic-selenium-webdriver.html#2)
* [Example: Fetch cell value of a particular row and column of the Dynamic Table](https://www.guru99.com/handling-dynamic-selenium-webdriver.html#3)
* [Example: Get Maximum of all the Values in a Column of Dynamic Table](https://www.guru99.com/handling-dynamic-selenium-webdriver.html#4)
* [Example: Get all the values of a Dynamic Table](https://www.guru99.com/handling-dynamic-selenium-webdriver.html#5)

**Using X-Path to Locate Web Table Elements**

Before we locate web element, first let's understands-

**What is a web element?**

Web elements are nothing but HTML elements like textbox, dropdowns radio buttons, submit buttons, etc. These HTML elements are written with **start** tag and ends with an **end** tag.

For Example,

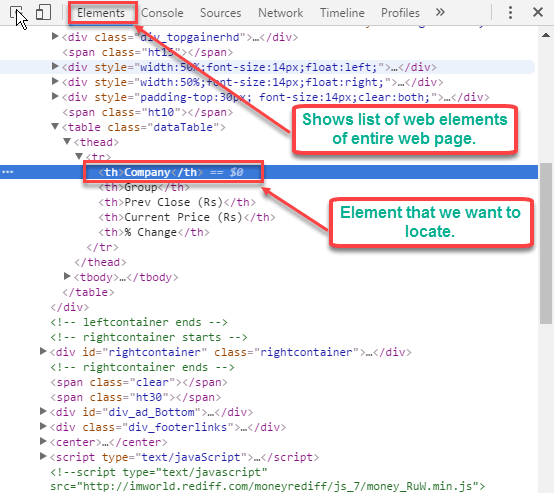
**<p>** My First HTML Document**</p>.**

Steps for getting X-path of web element that we want to locate.

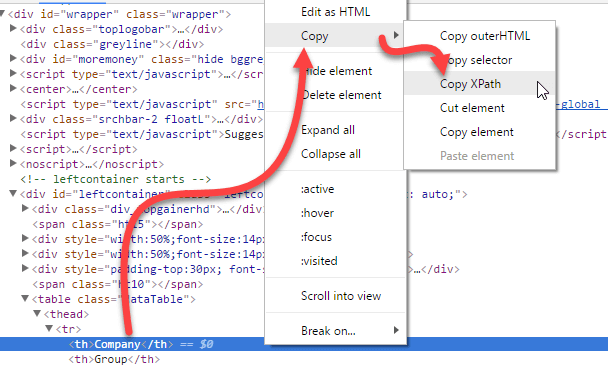
**Step 1)**In Chrome, Go to <http://money.rediff.com/gainers/bsc/daily/groupa>



**Step 2)**Right click on web element whose x-path is to be fetched. In our case, right click on "Company" Select Inspect option. The following screen will be shown -



**Step 3)** Right Click on highlighted web element > Select Copy -> Copy x-path option.



**Step 4)** Use the copied X-path "//\*[@id="leftcontainer"]/table/thead/tr/th [1]" in Selenium WebDriver to locate the element.

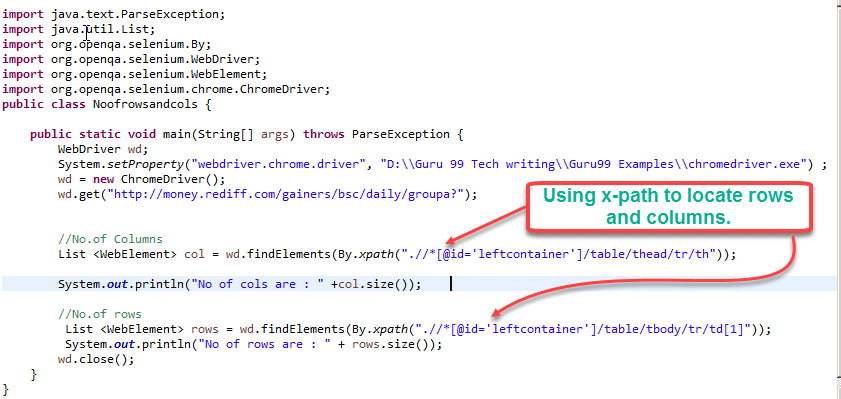
**Example: Fetch number of rows and columns from Dynamic WebTable**

When the table is dynamic in nature, we cannot predict its number of rows and columns.

Using Selenium web driver, we can find

* Number of Rows and columns of web table
* X row or Y column's data.

Below is program for fetching total number of rows and columns of web table.



import java.text.ParseException;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Noofrowsandcols {

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

WebDriver wd;

System.setProperty("webdriver.chrome.driver","G://chromedriver.exe");

wd= new ChromeDriver();

wd.get("http://money.rediff.com/gainers/bsc/dailygroupa?");

//No.of Columns

List col = wd.findElements(By.xpath(".//\*[@id=\"leftcontainer\"]/table/thead/tr/th"));

System.out.println("No of cols are : " +col.size());

//No.of rows

List rows = wd.findElements(By.xpath(".//\*[@id='leftcontainer']/table/tbody/tr/td[1]"));

System.out.println("No of rows are : " + rows.size());

wd.close();

}

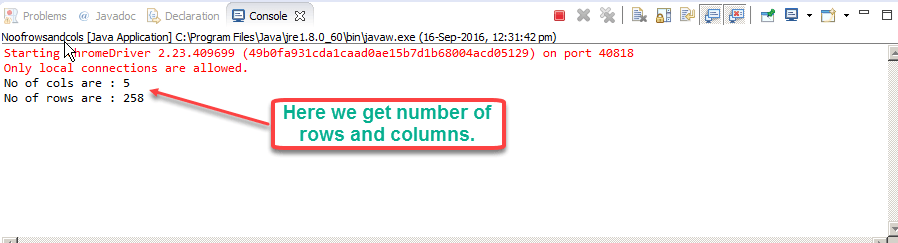
}

**Code Explanation:**

* Here we have first declared Web Driver object "wd" &initialized it to chrome driver.
* We use List <WebElement> to total number of columns in "col".
* findElements commands returns a list of ALL the elements matching the specified locator
* using findElements and X-path //\*[@id=\"leftcontainer\"]/table/thead/tr/th we get all the columns
* Similarly, we repeat the process for rows.

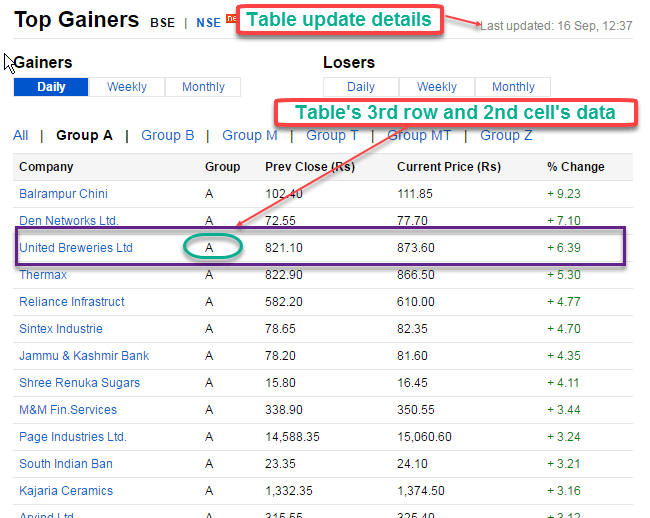
.

**Output:**

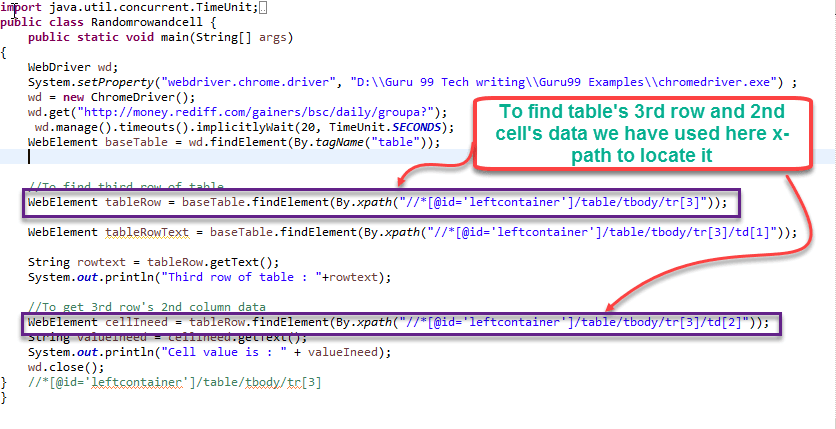


**Example: Fetch cell value of a particular row and column of the Dynamic Table**

Let's assume we need 3rd row of the table and its second cell's data. See the table below-



In above table, data is regularly updated after some span of time. The data you try retrieve will be different from the above screenshot. However, the code remains the same. Here is sample program to get the 3rd row and 2nd column's data.



import java.text.ParseException;

import java.util.List;

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 RowandCell {

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

WebDriver wd;

System.setProperty("webdriver.chrome.driver","G://chromedriver.exe");

wd= new ChromeDriver();

wd.get("http://money.rediff.com/gainers/bsc/daily/groupa?");

wd.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

WebElement baseTable = wd.findElement(By.tagName("table"));

//To find third row of table

WebElement tableRow = baseTable.findElement(By.xpath("//\*[@id=\"leftcontainer\"]/table/tbody/tr[3]"));

String rowtext = tableRow.getText();

System.out.println("Third row of table : "+rowtext);

//to get 3rd row's 2nd column data

WebElement cellIneed = tableRow.findElement(By.xpath("//\*[@id=\"leftcontainer\"]/table/tbody/tr[3]/td[2]"));

String valueIneed = cellIneed.getText();

System.out.println("Cell value is : " + valueIneed);

wd.close();

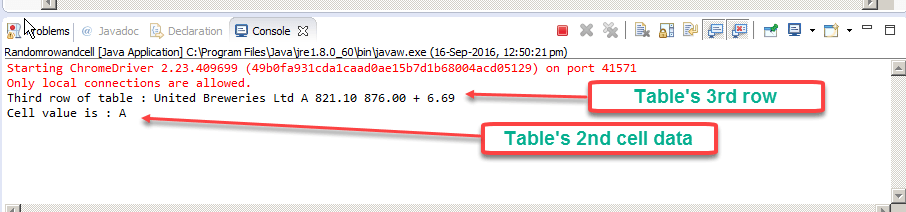
}

}

**Code Explanation:**

* Table is located using locator property "tagname".
* Using[XPath](https://www.guru99.com/xpath-selenium.html)"//\*[@id=\"leftcontainer\"]/table/tbody/tr[3]" find the 3rd row and gets its text using getText () function
* Using Xpath "//\*[@id=\"leftcontainer\"]/table/tbody/tr[3]/td[2]" find the 2nd cell in 3rdrow and gets its text using getText () function

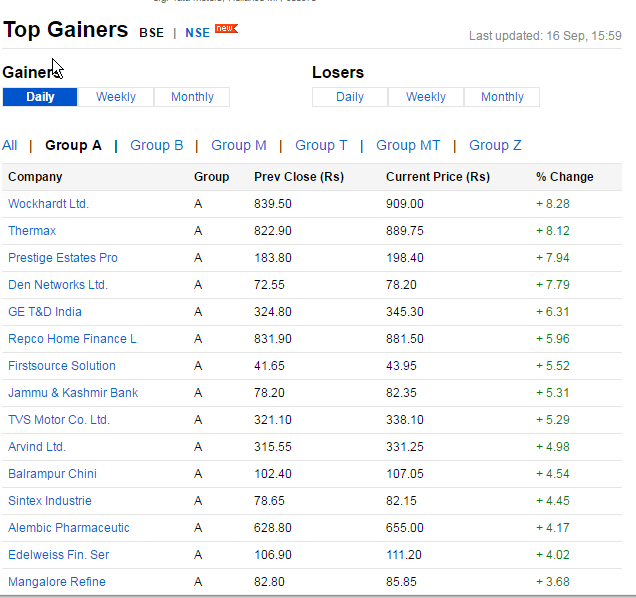
**Output**:



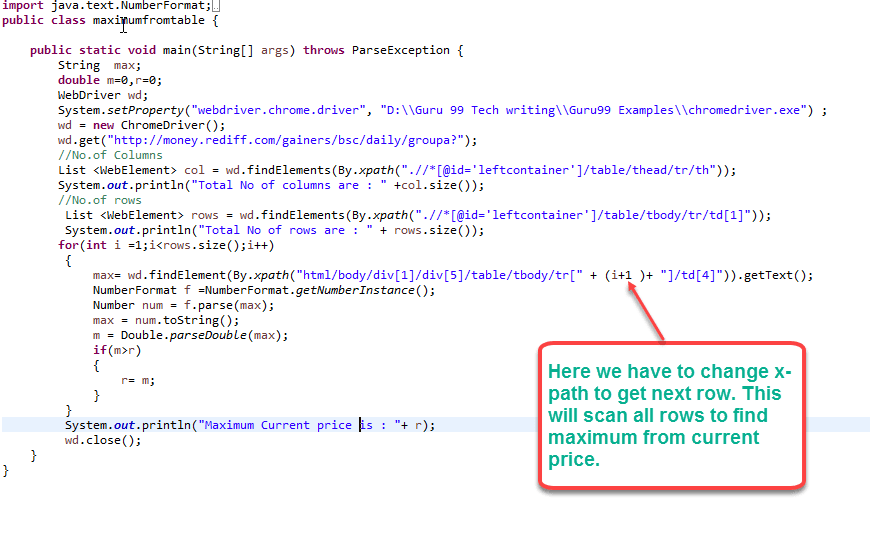
**Example: Get Maximum of all the Values in a Column of Dynamic Table**

In this example, we will get the maximum of all values in a particular column.

Refer the following table -



Here is the code



import java.text.ParseException;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.text.NumberFormat;

public class MaxFromTable {

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

WebDriver wd;

System.setProperty("webdriver.chrome.driver","G://chromedriver.exe");

wd= new ChromeDriver();

wd.get("http://money.rediff.com/gainers/bsc/daily/groupa?");

String max;

double m=0,r=0;

//No. of Columns

List col = wd.findElements(By.xpath(".//\*[@id='leftcontainer']/table/thead/tr/th"));

System.out.println("Total No of columns are : " +col.size());

//No.of rows

List rows = wd.findElements(By.xpath (".//\*[@id='leftcontainer']/table/tbody/tr/td[1]"));

System.out.println("Total No of rows are : " + rows.size());

for (int i =1;i<rows.size();i++)

{

max= wd.findElement(By.xpath("html/body/div[1]/div[5]/table/tbody/tr[" + (i+1)+ "]/td[4]")).getText();

NumberFormat f =NumberFormat.getNumberInstance();

Number num = f.parse(max);

max = num.toString();

m = Double.parseDouble(max);

if(m>r)

{

r=m;

}

}

System.out.println("Maximum current price is : "+ r);

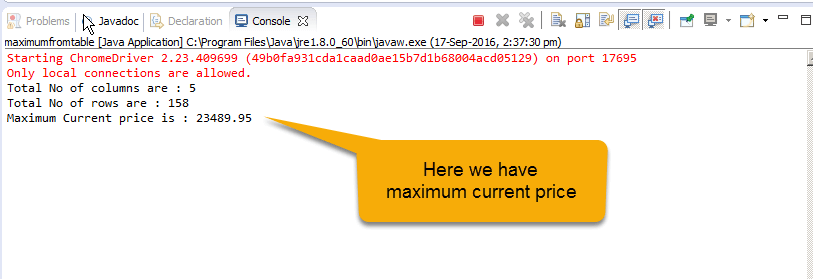
}

}

**Code Explanation:**

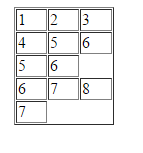
* Using chrome driver we locate the web table and get total number of row using XPath ".//\*[@id='leftcontainer']/table/tbody/tr/td[1]"
* Using for loop, we iterate through total number of rows and fetch values one by one. To get next row we use (i+1) in XPath
* We compare old value with new value and maximum value is printed at the end of for loop

**OutPut**



**Example: Get all the values of a Dynamic Table**

Consider the following table <http://demo.guru99.com/test/table.html>

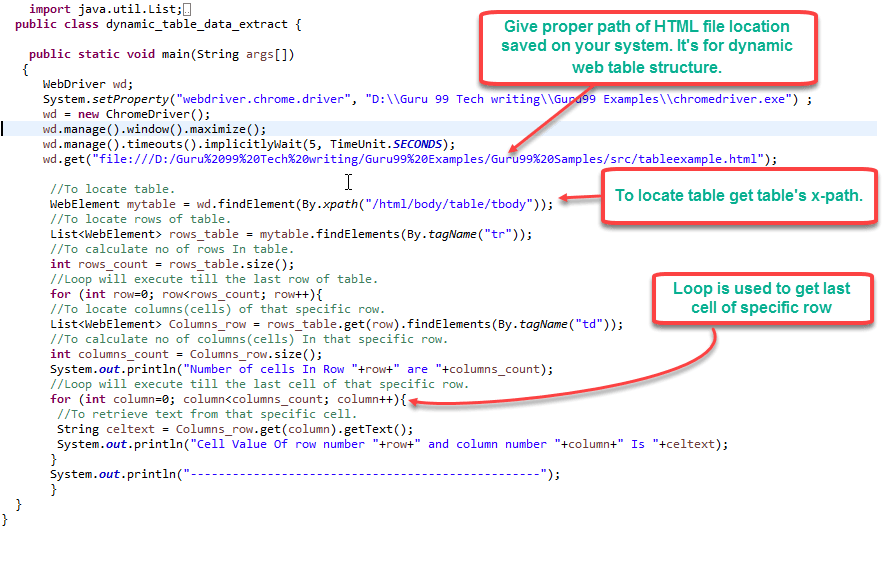


The number of columns for each row is different.

Here row number 1, 2 and 4 have 3 cells, and row number 3 has 2 cells, and row number 5 has 1 cell.

We need to get values of all the cells

**Here is the code:**



import java.text.ParseException;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.chrome.ChromeDriver;

public class NofRowsColmns {

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

WebDriver wd;

System.setProperty("webdriver.chrome.driver","G://chromedriver.exe");

wd = new ChromeDriver();

wd.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

wd.get("http://demo.guru99.com/test/table.html");

//To locate table.

WebElement mytable = wd.findElement(By.xpath("/html/body/table/tbody"));

//To locate rows of table.

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

//To calculate 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 < rows\_count; row++) {

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

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

//To calculate 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 < columns\_count; column++) {

// To 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);

}

System.out.println("-------------------------------------------------- ");

}

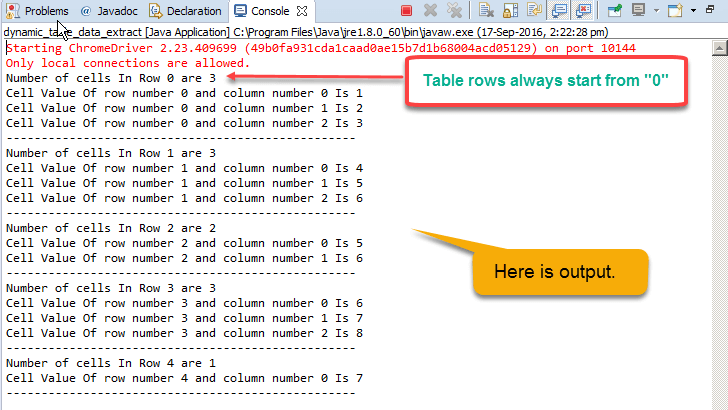
}

}

**Code Explanation:**

* rows\_count gives the total number of rows
* for each row we get the total number of columns using rows\_table.get(row).findElements(By.tagName("td"));
* We iterate through each column and of each row and fetch values.

**Output**:



**Summary**

* Static web tables are consistent in nature. i.e. they do have fixed number of rows as well as Cell data.
* Dynamic web tables are inconsistent i.e. they do not have fixed number of rows and cells data.
* Using selenium web driver, we can handle dynamic web tables easily.
* Selenium Webdriver allows us to access dynamic web tables by their X-path

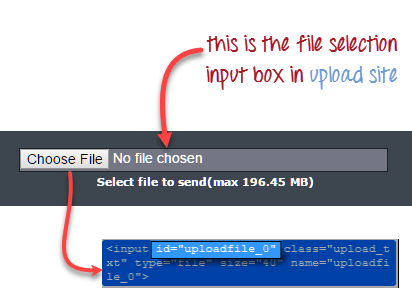
**6. How to Upload & Download a File using Selenium Webdriver**

In this tutorial, we will learn How to deal with file uploads and downloads.

**Uploading Files**

For this section, we will use <http://demo.guru99.com/test/upload/> as our test application. This site easily allows any visitor to upload files without requiring them to sign up.

**Uploading files in WebDriver is done by simply using the sendKeys() method on the file-select input field to enter the path to the file to be uploaded.**



Let's say we wish to upload the file "C:\newhtml.html". Our WebDriver code should be like the one shown below.

package newproject;

import org.openqa.selenium.\*;

import org.openqa.selenium.firefox.FirefoxDriver;

public class PG9 {

public static void main(String[] args) {

System.setProperty("webdriver.firefox.marionette","C:\\geckodriver.exe");

String baseUrl = "http://demo.guru99.com/test/upload/";

WebDriver driver = new FirefoxDriver();

driver.get(baseUrl);

WebElement uploadElement = driver.findElement(By.id("uploadfile\_0"));

// enter the file path onto the file-selection input field

uploadElement.sendKeys("C:\\newhtml.html");

// check the "I accept the terms of service" check box

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

// click the "UploadFile" button

driver.findElement(By.name("send")).click();

}

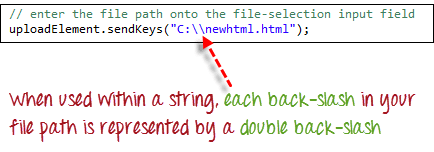
}

After running this script, you should be able to upload the file successfully and you should get a message similar to this.



Remember following two things when uploading files in WebDriver

1. There is no need to simulate the clicking of the "Browse" button. WebDriver automatically enters the file path onto the file-selection text box of the <input type="file"> element
2. When setting the file path in your Java IDE, use the proper escape character for the back-slash.



**Downloading Files**

**WebDriver has no capability to access the Download dialog boxes**presented by browsers when you click on a download link or button. However, we can bypass these dialog boxes using a separate program called "wget".

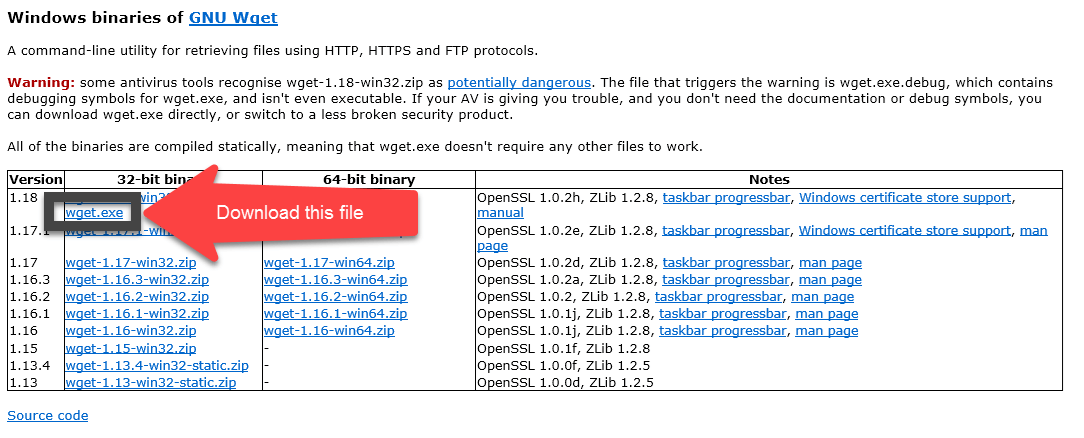
**What is Wget?**

**Wget is a small and easy-to-use command-line program used to automate downloads**. Basically, we will access Wget from our WebDriver script to perform the download process.

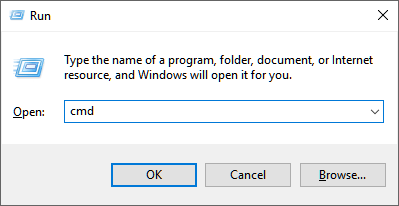
**Setting up Wget**

**Step 1:**In your C Drive, create a new folder and name it as "Wget".

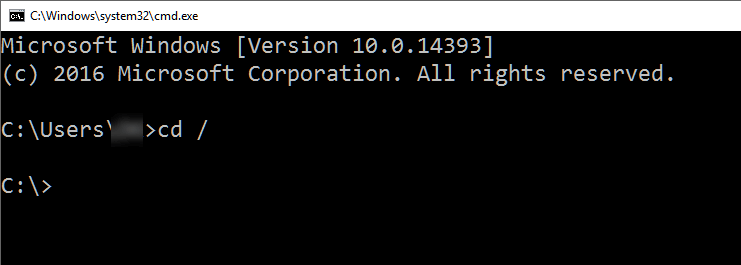
Download wget.exe [from here](https://eternallybored.org/misc/wget/) and Place it in the Wget folder you created from the step above.



**Step 2:**Open Run by pressing windows key + "R" ; type in "cmd & click ok

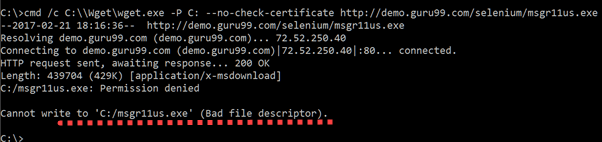


Type in the command "cd /" to move to the root directory



**Step 3:**Type in the command to check whether the given setup is working

cmd /c C:\\Wget\\wget.exe -P C: --no-check-certificate http://demo.guru99.com/selenium/msgr11us.exe

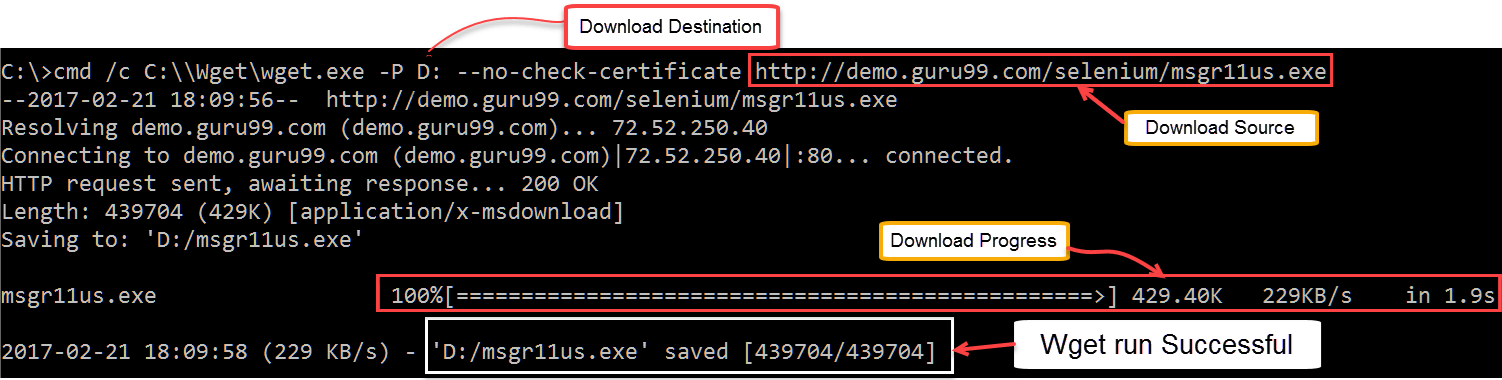


There seems to be an issue writing into C drive.

**Step 4:**You need to debug the wget errors in command line before you execute the code using Selenium Webdriver. These errors will persist in Eclipse and the error messages will not be as informative. Best to first get wget working using command line. If it works in command line it will definitely work in Eclipse.

In our example, as show in step 3, there is a problem writing into C drive. Let's change the download location to D drive and check results.

cmd /c C:\\Wget\\wget.exe -P D: --no-check-certificate http://demo.guru99.com/selenium/msgr11us.exe

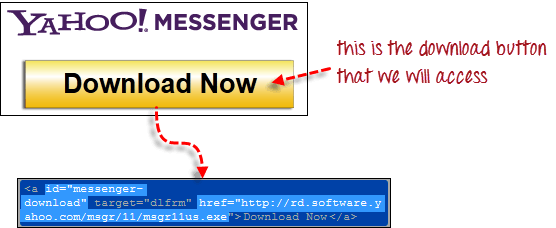


Messenger was downloaded successfully.

Before you proceed further don't forget to delete the downloaded file

**Using WebDriver and Wget**

In the following example, we will use WebDriver and wget to download a popular chat software called Yahoo Messenger. Our base URL shall be <http://demo.guru99.com/test/yahoo.html>.



**Step 1**

Import the "java.io.IOException" package because we will have to catch an IOException later in Step 4.



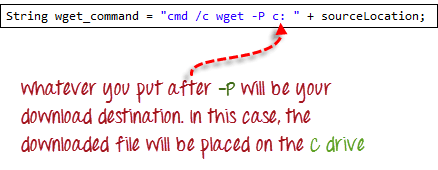
**Step 2**

Use getAttribute() to obtain the "href" value of the download link and save it as a String variable. In this case, we named the variable as "sourceLocation".



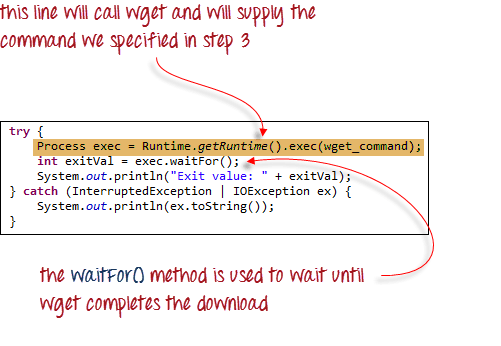
**Step 3**

Set-up the syntax for wget using the following command.



**Step 4**

Initiate the download process by calling wget from our WebDriver code.



To sum it all up, your WebDriver code could look like the one shown below.

package newproject;

import java.io.IOException;

import org.openqa.selenium.\*;

import org.openqa.selenium.firefox.FirefoxDriver;

public class PG8 {

public static void main(String[] args) {

System.setProperty("webdriver.firefox.marionette","C:\\geckodriver.exe");

String baseUrl = "http://demo.guru99.com/test/yahoo.html";

WebDriver driver = new FirefoxDriver();

driver.get(baseUrl);

WebElement downloadButton = driver.findElement(By

.id("messenger-download"));

String sourceLocation = downloadButton.getAttribute("href");

String wget\_command = "cmd /c C:\\Wget\\wget.exe -P D: --no-check-certificate " + sourceLocation;

try {

Process exec = Runtime.getRuntime().exec(wget\_command);

int exitVal = exec.waitFor();

System.out.println("Exit value: " + exitVal);

} catch (InterruptedException | IOException ex) {

System.out.println(ex.toString());

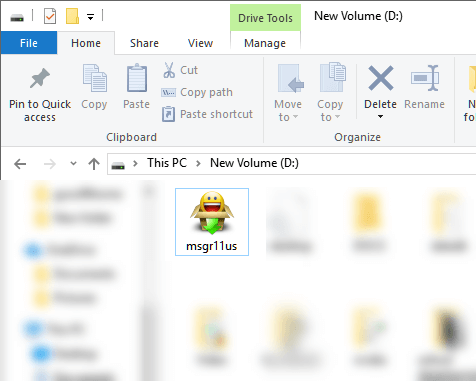
}

driver.close();

}

}

After executing this code, check your D drive and verify that the Yahoo Messenger installer was successfully downloaded there.



**Summary**

* Uploading files in WebDriver is done by simply using the sendKeys() method on the file-select input field to enter the path to the file to be uploaded.
* WebDriver cannot automate downloading of files on its own.
* The easiest way to download files using WebDriver is to use Wget.