Vinay2ml@gmail.com

Selenium: Webdriver:

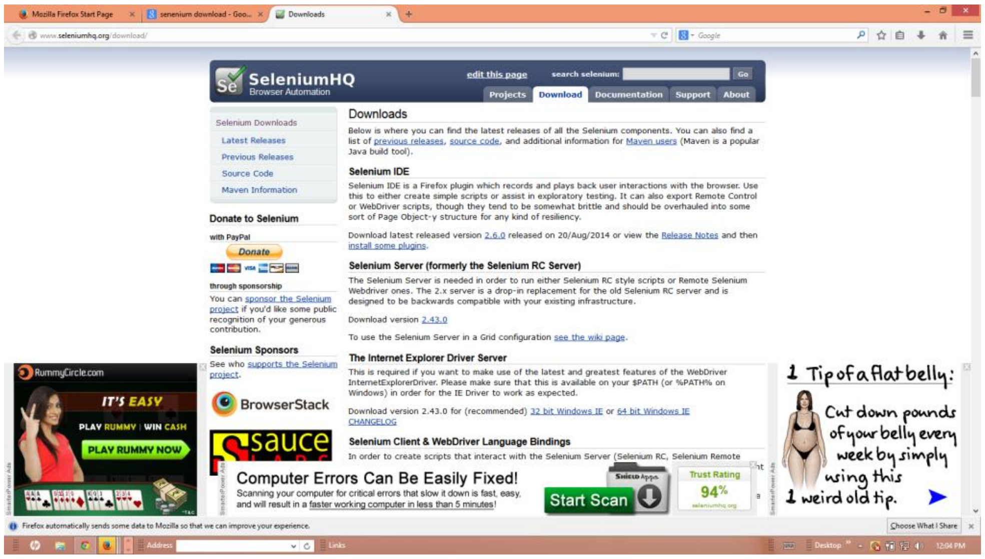
It is a free and open source web application automation tool, which performs the action on the application by calling native methods of the browser.

Installing selenium web driver: or setup the web driver

1) Required software's:
2) JDK(Java development kit) latest version 1.7 or 1.8
3) Eclipse IDE
4) Mozilla Firefox browser
5) Selenium jar file

Download the selenium jar file steps:

1) Open browser and click this link :



Downloads

Below is where you can find the latest releases of all the Selenium components. You can also find a list of previous releases, source code, and additional information for Maven users (Maven is a popular Java build tool).

Selenium IDE

Selenium IDE is a Firefox plugin which records and plays back user interactions with the browser. Use this to either create simple scripts or assist in exploratory testing. It can also export Remote Control or WebDriver scripts, though they tend to be somewhat brittle and should be overhauled into some sort of Page Object-y structure for any kind of resiliency.

Download latest released version 2.6.0 released on 20/Aug/2014 or view the Release Notes and then install some plugins.

Selenium Server (formerly the Selenium RC Server)

The Selenium Server is needed in order to run either Selenium RC style scripts or Remote Selenium Webdriver ones. The 2.x server is a drop-in replacement for the old Selenium RC server and is designed to be backwards compatible with your existing infrastructure.

Download version 2.43.0

To use the Selenium Server in a Grid configuration see the wiki page.

The Internet Explorer Driver Server

This is required if you want to make use of the latest and greatest features of the WebDriver InternetExplorerDriver. Please make sure that this is available on your $PATH (or %PATH% on Windows) in order for the IE Driver to work as expected.

Download version 2.43.0 for (recommended) 32 bit Windows IE or 64 bit Windows IE
CHANGELOG

Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Selenium RC, Selenium Remote

Donate to Selenium

with PayPal

Donate

through sponsorship

You can sponsor the Selenium project if you'd like some public recognition of your generous contribution.

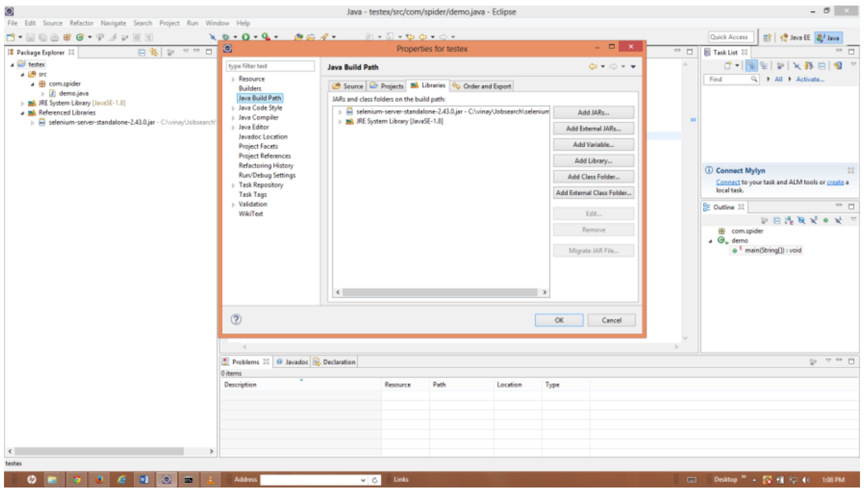
Selenium Sponsors

See who supports the Selenium project.

BrowserStack
sauce
RummyCircle.com
IT'S EASY
PLAY RUMMY | WIN CASH
PLAY RUMMY NOW
Computer Errors Can Be Easily Fixed!
Scanning your computer for critical errors that slow it down is fast, easy, and will result in a faster working computer in less than 5 minutes!
Start Scan
Trust Rating
94%
1 Tip of a flat belly:
Cut down pounds of your belly every week by simply using this 1 weird old tip.
Firefox automatically sends some data to Mozilla so that we can improve your experience.
Choose What I Share
Address
Links
Desktop
12:04 PM

2) Click download version on selenium server section :
   2.43.0 And save into respective location

3) Open Eclipse create a java project


Java - testex/src/com/spider/demo.java - Eclipse
Properties for testex
Java Build Path
Source
Projects
Libraries
Order and Export
JARs and class folders on the build path:
selenium-server-standalone-2.43.0.jar - C:\vinay\Jobsearch\selenium
JRE System Library [JavaSE-1.8]
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
OK
Cancel

Select->properties->Java build path→Libraries -> Add external Jar files
And select the selenium jar file and click ok.

Selenium supports following languages:

1) Java
2) C#
3) Python
4) Java script
5) Perl
6) Php

Ex:

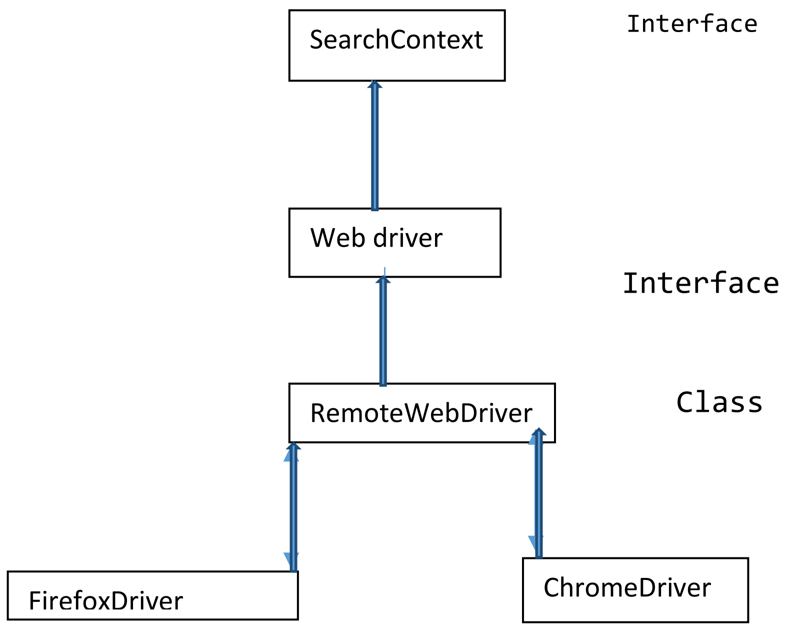
```
package com.spider;

import org.openqa.selenium.firefox.FirefoxDriver;

publicclass demo {

	publicstaticvoid main(String[] args) {
		// TODO Auto-generated method stub
		FirefoxDriver f =new FirefoxDriver();
		f.close();
	}

}
```

Advantages:

1) Free :selenium can be used for commercial purpose also without purchasing the licensee
2) Open source: you can view and download and customize the source code of selenium itself.
3) Web application: any application which is opened browser and automated using selenium
4) Ex: Google.com, Facebook.com

5) Automation tool: it is software application which is used to test another application automatically
6) Native method: The methods which are available in the browser and which are exposed by browser developers are called as native methods. Ex : close() Firefox

Architecture of selenium web driver:



Search context: is a super most interface which is extended by webDriver interface.

RemoteWebdriver: class implements all the abstract methods of the both interface. The browser specific class such as Firefox, chrome driver, internet explorer driver, safari driver, extends Remote webdriver class.

We do browser compatibility testing that is use a browser such as Firefox, while writing the automation script but will run it on different browser. In order to do this we will use runtime-polymorphism concept to achieve this using up casting in selenium, we always up cast the browser object, web driver interface as below.

```
Ex: Webdriver driver = new FirefoxDriver ();
```

1) How do you close the browser?
   Webdriver driver =new FirefoxDriver ();
   Driver.close();

2) How do you close the browser without using close() browser ?
   WebDriver driver = new FirefoxDriver ();
   Driver.quit()

3) How do you open the web page ?
   webDriver driver = new FirefoxDriver();
   driver.get("http://www.google.com");
   driver.close();

   We should specify complete URL of the application starting from the protocol and get Method make the selenium to wait till the webpage is completely loaded.

4) How do you open the webpage without using the get method ()?
   Webdriver driver = new FirefoxDriver ();
   driver.navigate.to ("http://www.bing.com");
   driver. Close ()

5) How do you click on back button in browser web page.

```
WebDriver driver =new FirefoxDriver();
driver.get("http:\\www.google.com");
driver.navigate().to("http:\\www.bing.com");
driver.navigate().to("http:\\www.gmail.com");
driver.navigate().back();
String utl = driver.getCurrentUrl();
if(utl.contains("bing")){
	System.out.println(utl);
	System.out.printf("Back page %s loaded successfully",utl);
}else{
	System.out.println("bing page is not loaded");
}
driver.close();
```

6) How do you click on forward button in browser ?

```
publicclass pythonfile {
publicstaticvoid main(String[] args) {
	WebDriver driver = new FirefoxDriver();
	driver.get("http:\\www.google.com");
	driver.navigate().to("http:\\www.gmail.com");
	driver.navigate().back();
	driver.navigate().forward();
	driver.close();
```

```
      }
      7) How do you click on forward button in browser web page.

publicclass forwardback {

            publicstaticvoid main(String[] args)
             {
            WebDriver driver =new FirefoxDriver();
            driver.get("http:\\wwww.bing.com");
            driver.navigate().to("http:\\www.google.com");
            driver.navigate().back();
            driver.navigate().forward();
            if(driver.getCurrentUrl().contains("google.co.in")){
                  System.out.println("Google.co.in loaded successfully ");

            }else
            {
                  System.out.println("Google.co.in not laoded ");
            }
            driver.close();
      }

         }
      8)  How do you refresh the web page?

      publicclass refreshpage {
      publicstaticvoid main(String[] args) {
            WebDriver driver = new FirefoxDriver() ;
            driver.get("file:///C:/MPS/HTML_Scripts/Client_9.2_Devices.htm");
            driver.navigate().refresh();
            driver.close();

      }

      }

      9) How do you maximize the web page ?

      publicclass maxmizepage {
      publicstaticvoid main(String[] args) {
            WebDriver driver = new FirefoxDriver();
            driver.get("file:///C:/MPS/HTML_Scripts/Client_9.2_Devices.htm");
            driver.navigate().refresh();   //refresh the page
            driver.manage().window().maximize() ;
            driver.close();


      }

      }
```

10) Write a script open Google in web page and verify that google page is loaded ?

```
publicclass gettitilepage {

publicstaticvoid main(String[] args) {
        WebDriver driver = new FirefoxDriver();
        driver.get("http:\\www.google.com");
        String title = driver.getTitle();
        if(title.equals("Google")){
                System.out.println("Goole title is present");
        }else {
                System.out.println("Google title is not present");
        }
        driver.close() ;
}

}
```

String comparison if case sensitive to handle such scenarios use Equalignorecase().
If the value keep changing then we can use contains() function.

```
publicclass gettitilepageignorecase {


publicstaticvoid main(String[] args) {
        WebDriver driver = new FirefoxDriver();
        driver.get("http:\\www.google.com");
        if(driver.getTitle().equalsIgnoreCase("google")){
                System.out.println("Google titile present");
        }
        else{
                System.out.println("Google title is not present");
        }
        driver.close();
}

}
```

11) Write a code to get the url present in the address bar ?

```
        publicstaticvoid main(String[] args) {
        WebDriver driver = new FirefoxDriver();
        driver.get("http:\\www.google.com");
        String url =driver.getCurrentUrl();
        System.out.println(url);
        driver.close();

        }
```

Assignment:

1) Write a script open Google.com verify that is navigated to Google.co.in

```
publicclass geturl {

publicstaticvoid main(String[] args) {
       WebDriver driver = new FirefoxDriver();
       driver.get("http:\\www.google.com");
       String url =driver.getCurrentUrl();
       System.out.println(url);
       if(url.contains("google.co.in")){
              System.out.println("Google.com navigated to goodle.co.in
       successfully");
       }else {
              System.out.println("Google.com is not navigated to google.co.in
       ");
       }
       driver.close();

       }

       }
```

HTML: (hypertext markup Language).
Basically use to develop web pages in selenium, before performing any action on the elements (button, textboxes, and links), it is to identify that element uniquely in order to do this, it uses characteristics of those elements, which is given by the application developer using HTML.

Characteristics can also be called as attributes or properties.
HTML is not case sensitive
We can use notepad itself to write the html code but by saving the code using .html as extension.

Use predefined key words with in angle bracket <html> these are called html tags.
Use Forward '/' to end the tag ending all the tag is mandatory </html>

Ex:

```
<html>
<title>Qspider</title>
<body>
           Welcome !
</body>
      </html>
```

Important Web elements:

1) Toolbox   UN: <input type =’text’ value = “abc”>
2) PasswordFiled   PW: <input type =”password” value =”xyz”>
3) Checkbox   <input type = “checkbox”>
4) Button   <input type = “button” value = “ok”
5) Radiobutton <input type =”radio”>
6) Filebrowser <input type =”file”>
7) Image   <img src =”log.png”>
8) List items

```
<select>
<Option> idly </option>
<Option>vada</option>
<Option>dosa</option>
<Option>pulhogra</option>
</select>
```

9) Multi select code <select multiple = “true”>
10) Hyper link <a href =”http://www.google.com”, title =”click here”>Qspider</a>

Create a table:

```
<html>
<body>
     <table border =”1”>
<tbody>
     <tr>
          <th>snio</th>
          <th>subject</th>
          <th>Apply</th>
</tr>
     <tr>
          <td>1</td>
          <td>Java</td>
          <td>input type =”checkbox”></td>
     </tr>
     </tbody>
</table>
     </body>
     </html>
```

Component of web element:

```
<Html tags>
<html>
<body>
<title>
```

<input> , <img>, <select> , <option><a><link>, <br>
<table> , <tbody>, <tr><td><th>, <head>, <script>,
<form> , <span>, <li>, <ol>, <ul><iframe>, <div>

Any pair of words separated by "=" is called as attribute also called as, property name and property value.
The attribute will be present after the html tag till "<" symbol.
Ex:
Type ="text"
Value "abc"
Src ="lon.png"
Multiple ="true"
Tittle ="checkbox"
Boder ="1"
Id ="username"
Class ="textfiled"
Name ="username"
Placeholder ="username"

Any word which is present after the greater than symbol till the end of respective html tag is called as text of the element.

<div class ="lable" >Tasks </div> } web element
(div) tag -> gettagname();
Getattribute(property name)
Gettext()

Locator: Selenium identify the element using locators and there are 8 types of locator are available.

1)tagName   - tag
2) id , name, className are called attribute
3) linkText, parstialLinkText, are called Text
4) CssSelector , xpath are called Expression

Note: all the locator are present in "By" class as static methods, they take as string as argument and return on object of type "By" this object is used in find element.
findElement() and findElemets();
to search element in the webpage, if search is successful element is found and get return object of type webelement.

Ex:
<html>

```
<body>
< a href =”http://www.google/com” id =”a1” name =”n1” class
=”c1”>Qspider</a>
</body>
</html>
```

1)Selenium Script:To click on the link on the sample web page using “tagName()” as locator.

```
publicclass tagNameclick {
publicstaticvoid main(String[] args) {
WebDriver driver = new FirefoxDriver();
driver.get("file:///C:/selenium/HTMLfiles/Mypage.html");
driver.findElement(By.tagName("a")).click();
driver.close();
}
}
```

2)Selenium Script: To click on the link on the sample web page using attribute “id()“as locator.

```
publicclass attribute_id {
publicstaticvoid main(String[] args) {
WebDriver driver = new FirefoxDriver();
driver.get("file:///C:/selenium/HTMLfiles/Mypage.html");
driver.findElement(By.id("a1")).click();
driver.close();
}
}
```

4) Selenium Script: To click on the link on the sample web page using attribute “className()” as locator.

```
publicclass attribute_class {

    publicstaticvoid main(String[] args) {
    WebDriver driver = new FirefoxDriver();
    driver.get("file:///C:/selenium/HTMLfiles/Mypage.html");
    driver.findElement(By.className("c1")).click();


    }

    }
```

5) Selenium Script: To click on the link on the sample web page using text property “linkText()” as locator.

```
publicclass testfindelement {
      publicstaticvoid main(String[] args) {
            // TODO Auto-generated method stub
            WebDriver driver = new FirefoxDriver();
            driver.get("file:///C:/selenium/HTMLfiles/Mypage.html");
            driver.findElement(By.linkText("Qspider")).click();

       }

}
```

6) Selenium Script: To click on the link on the sample web page using text "PatrialLinkText() property as locator.

```
publicclass partiallinktext {

publicstaticvoid main(String[] args) {
      WebDriver driver = new FirefoxDriver() ;
      driver.get("file:///C:/selenium/HTMLfiles/Mypage.html");
      driver.findElement(By.partialLinkText("spider")).click();
      driver.close();
}

}
```

If specified locator is not matching with any of the element then findElement() method will throw Exception "NoSuchElementException"

Error ex:

```
Exception in thread "main" org.openqa.selenium.NoSuchElementException: Unable to
locate element: {"method":"partial link text","selector":"spider12"}
```

If the specified locator is matching with more than one element(duplicate) then findElement() method returns first matching web Element.

cssSelector: sometime we cannot use the first 6 type of locator(tagName(), id(), name(), className(),linkText(),paritialLinkText()) because it may be not supported. Or it may not be given by developer or it may be duplicated.

Ex:

```
<html>
<body>
          UN: <input type ="text" ><br>
          PW: <input type ="Password">
</body>
</html>
```

In the above webpage to identify password filed, we cannot use the linkText() or partialLinkText() as those not supported for the password filed. We can't use the id() , name() , className(), as it not given by the developer , we can't use tagName() as it is duplicate. In such scenario we can use the cssSelector,(css(Cascaded style sheet)) with following syntax.

Htmltag[property ="value"]

```
Ex: input[type ='text']
    Input[type ='password']
```

```
publicclass cssselectorexpression {
publicstaticvoid main(String[] args) {
	WebDriver driver = new FirefoxDriver();
	driver.get("file:///C:/selenium/HTMLfiles/cssselectorex.html") ;
	driver.findElement(By.cssSelector("input[type='text']")).sendKeys("vinay");
	driver.findElement(By.cssSelector("input[type='password']")).sendKeys("Jugga")
;
	driver.close();
	}
}
```

Note: we can also use in Double cote in the “cssSelector(“input[type =\”password\”])” but it we should be escape it using “\” backward slah.

```
  Ex: input[id=’username’]
      Input[class=’textField’]
       Input[name=’username’]
       Input[placeholder=’username’]
```

1) What are the attributes supported by the locator
   Id(),name(), class().

2) Then how to use other attribute to identify the elements
   We can use cssSelector() and xpath

Xpath: sometime even cssSelector() also can be duplicate.

```
	Ex: <html>
	<body>
		FN:<input type=”text”><br>
		LN:<input type =”text”>
	</body>
	</html>
```

In the above webpage both first name and last name as same cssexpression, hence we can’t enter the lastname in such cases we use Xpath.

Xpath: it is path of the element in html tree.

1) While writing the xpath expression the specify the path using “/”far wordslash.
2) First forward ‘/’ slash represent beginning of html tree, it is called as root.
3) After every “/” slash we must specify html tag of immediate child node only.
4) If there are duplicate elements are there then we can use index , it is with in square bracket “[]” index start

Index is sequential no generated for duplicate sibling only. (Same tag name under same
	Parent element.

```
	 Body
	   |
	Input
	   |
	 A
	 |
       ----→Div1 → input1
                → Input2
                → A
        ----→Div2 →input 1
                  ->input 2
```

In order to view the source code of the element it is called as inspecting the element use Firebug, which is Add-on for firebox browser. In order to install it in the firefox browser.
Tools->Add-on

```
      Tree structure:
                  Html
                  ↓----→body
                  |--→div
                        |---→input A
                        |--→input B
                  |
                  |____div
                        |--→input c
                        |-→input D
```

Absolute xpath: specify the complete path of the element is called as absolute xpath.

Absolute Xpath	Elements
/html/body/div/input	A, B, C, D
/html/body/div[1]/input	A, B
/html/body/div[1]/input[1]	A
/html/body/div[1]/input[2]	B

Note: Navigating from parent node to the child node is called as forward traversing

Relative xpath: The absolute path is very lengthy instead of this we can use releative xpath whith help of "//"

"/" represent immediate child
"//" represent any child descended

Relative xpath:	Element
//input	A, B, C, D
//input[1]	A, C
//input[2]	B, D
//div[1]/input	A, B
//div[1]/input[1]	A
//div[1]/input[2]	B
//div[2]/input	C, D
//div[2]/input[1]	C
//div[2]/input[2]	D

What is difference between //a &

1) Write the program to logout from the application without using any of synchronization option?

```
publicclass withoutsync {

    publicstaticvoid main(String[] args) {
        WebDriver driver = new FirefoxDriver();
        driver.get("http://demo.actitime.com");
        driver.findElement(By.id("username")).sendKeys("admin");
        driver.findElement(By.name("pwd")).sendKeys("manager");
        driver.findElement(By.id("loginButton")).click() ;
        while(true)
        {
            try
            {
                driver.findElement(By.id("logoutLink")).click();
                break;
            }
            catch(NoSuchElementException e)
            {
            }

        }

    driver.close();
    }

}
```

2) Write a script login to the application without using click () method.

```
    publicclass withoutclick {

    publicstaticvoid main(String[] args) {
        WebDriver driver = new FirefoxDriver();
        driver.get("http://demo.actitime.com");
        driver.findElement(By.name("username")).sendKeys("admin");
        driver.findElement(By.name("pwd")).sendKeys("manager");
        driver.findElement(By.id("loginButton")).sendKeys(Keys.ENTER);
        driver.close();

    }

}
```

3) Write a script to change text present in the textbox

```
    publicclass changetext {
    publicstaticvoid main(String[] args) {
        WebDriver driver = new FirefoxDriver();
        driver.get("https://demo.vtiger.com/");
        WebElement un =driver.findElement(By.id("username"));
        un.clear();
        un.sendKeys("vinay");
        WebElement up =driver.findElement(By.id("password"));
        up.clear();
        up.sendKeys("Jugga");
        driver.close();

    }

    }
```

4) Write a script change the text present in the text box without using the clear() method ?

```
publicclass withoutclear
 {
publicstaticvoid main(String[] args)
{
        WebDriver driver =new FirefoxDriver();
        driver.get("https://demo.vtiger.com/");
        WebElement un =driver.findElement(By.id("username"));
        un.sendKeys(Keys.CONTROL,"a");
        un.sendKeys("manager");
        driver.close();

}

}
```

5) Write a script to copy and paste text one box into another text box

```
publicclass copypaste {

    publicstaticvoid main(String[] args) {
        WebDriver driver =new FirefoxDriver();
        driver.get("https://demo.vtiger.com/");
        //WebElement up =driver.findElement(By.id("password"));
        //up.sendKeys(Keys.CONTROL,"a");
        //up.sendKeys(Keys.CONTROL,"c");
        //WebElement un =driver.findElement(By.id("username"));
        //un.clear();
        //un.sendKeys(Keys.CONTROL,"v");

        WebElement up =driver.findElement(By.id("username"));
        up.sendKeys(Keys.CONTROL,"a");
        up.sendKeys(Keys.CONTROL,"c");
        WebElement un =driver.findElement(By.id("password"));
        un.clear();
        un.sendKeys(Keys.CONTROL,"v");
        driver.close();

    }

    }
```

1) Write a script to get text present in the text box.
Note: The text will be present inside an attribute called value, hence we should use getAttribute() method.

```
publicclass gettext {
	publicstaticvoid main(String[] args) {
		WebDriver driver = new FirefoxDriver();
		driver.get("https://demo.vtiger.com/");
			String text
	=driver.findElement(By.id("username")).getAttribute("value");
		System.out.println(text);

		String pass
=driver.findElement(By.id("password")).getAttribute("value");
		System.out.println(pass);
		driver.close();
	}

}
```

Note: Can we return the pass word is present in the password filed using selenium?
 Yes.

2) Write the script to get url of the link ?

```
	publicclass geturl {

	publicstaticvoid main(String[] args) {
		WebDriver driver = new FirefoxDriver();
		driver.get("https://demo.vtiger.com/");
		String url = driver.findElement(By.linkText("Vtiger
Website")).getAttribute("href");
		System.out.println(url);
		driver.close();
}
	}
```

3) Write a script to get tool tip text of the link from the filpkart.com

```
	publicclass tooltiptext {

	publicstaticvoid main(String[] args) {
		WebDriver driver = new FirefoxDriver();

	driver.get("http://www.flipkart.com/tablets/apple~brand/pr?sid=tyy,hry&otracke
r=hp_nmenu_sub_electronics_0_iPad");
		String str1 =driver.findElement(By.partialLinkText("Apple 16GB iPad Mini
with Wi-Fi")).getAttribute("title");
		System.out.println(str1);
		driver.close();


	}

	}
```

4) Write a script to get the price of IPad min present in the filpkart.com

5) Write a script to get phone no of Mumbai present in the irctc.com

```
        publicstaticvoid main(String[] args) {

        WebDriver driver = new FirefoxDriver() ;
        driver.get("http://www.irctc.com");

        String xp1 ="//label[text()='Mumbai']/..//label[text() =': 022-
22618067']" ;
        String text1 =driver.findElement(By.xpath(xp1)).getText();
        System.out.println(text1);

        String xp2 ="//label[text()='For Internet Ticketing Complaints
']/..//label[text() =': care@irctc.co.in']" ;
        String text2 = driver.findElement(By.xpath(xp2)).getText() ;
        System.out.println(text2);

        String text3 =driver.findElement(By.partialLinkText("INDIAN RAILWAY
CATERING")).getAttribute("title") ;
        System.out.println(text3);


    }
```

6) Write a script select the checkbox whether that is selected or not

```
    publicclass checkboxseleted {


    publicstaticvoid main(String[] args) {
    WebDriver driver = new FirefoxDriver();
    driver.get("http://demo.actitime.com");
    WebElement chebox1 = driver.findElement(By.id("keepLoggedInCheckBox"));
    chebox1.click();
    if(chebox1.isSelected()){
        System.out.println("Check box is selected");

    }
    else{
        System.out.println("check box is not selected");
    }
    driver.close();
    }


}
```

5) How do you verify whether the button is enabled or not isEnabled() method which returns Boolean value.

Ex:

```
publicclass buttonenabled {


publicstaticvoid main(String[] args) {
      WebDriver driver = new FirefoxDriver();
      driver.get("http://demo.actitime.com");
      WebElement btton = driver.findElement(By.id("loginButton")) ;
      if(btton.isEnabled()) {
            System.out.println("Button is enabled");
      }
      else
      {
            System.out.println("button is not enabled");
      }

}

}
```

6) Write a script to verify weather username text box is displayed or not in the login page.

```
publicstaticvoid main(String[] args) {

      WebDriver driver = new FirefoxDriver();
      driver.get("http://demo.actitime.com");
      WebElement enab1=driver.findElement(By.id("username"));
      if (enab1.isDisplayed())
      {
            System.out.println("checkbox displayed");
      }
      else
      {
            System.out.println("not displayed the checkbox");
      }
      driver.close();
}

}
```

Important Methods and web Element:

1) Click() ;
2) Clear() ;
3) sendKeys(character sequence…) ;
4) getAttributes(“property”) ;
5) getTitle();
6) isDisplayed();
7) isEnabled() ;
8) isSelected();
9) getLocation() return the coordinates x and y
10) getsize() return the width and height
11) getcssValue(String);
12) Submit();

Find Elements(findElements()):
In order to get multiple elements from the webpage we use findElements method which returns List<WebElement> , Generally we use xpath for this method.

Difference between findElement and findElements.

findElement()	findElements()
1) It returns WebElement	It returns List of WebElement
2) If locator is matching with n elements(multiple) then it returns first matching WebElement	If locator is matching with n elements it returns List of webElemetns with size n and index 0 to n-1
3) If locator is matching with one elements it returns matching WebElement	If the locator is matching with one element , it returns List of webElement with size 1 and index 0
4) If the locator is not matching with any of the element then it will throw NoSuchElementException	The locator is not matching with any of the element then also it returns list of webelement with size 0 and without any index, it also called as empty list.

We must import list from java.utl package and it is an interface.

Ex:

1) Count all the links present on the webpage and clicking on the first link .

```
publicclass countlinks {
publicstaticvoid main(String[] args) {
      WebDriver driver = new FirefoxDriver();
      driver.get("file:///C:/selenium/HTMLfiles/Mypage.html");
      List<WebElement> alllinks ;
      alllinks =driver.findElements(By.xpath("//a")) ;
      int linkcount= alllinks.size();
      System.out.println(linkcount);
      WebElement link = alllinks.get(2);
      link.click() ;
      driver.close();


}

}
```

2) Write a script to print text of the links which is present on the google.com except empty string

```
publicclass googlelinks {

publicstaticvoid main(String[] args) {
      WebDriver driver = new FirefoxDriver();
      driver.get("http:\\www.google.com") ;
      List <WebElement> alllinks =driver.findElements(By.xpath("//a"));
      int countlinks= alllinks.size();
```

```
            System.out.println(countlinks);
            int j =0;
            for(int i =0 ;i < countlinks ;i++){
                   WebElement link = alllinks.get(i);
                    String linktext = link.getText();
                  int linksize =linktext.length();
                  // System.out.println(linktext);

                  if(linksize >0){
                          j+=1 ;
                          System.out.println(linktext);

                    }

            }
             System.out.println(j);
            driver.close();

      }

      }
```

3) Write a script to select all the checkbox present on the webpage in GSMarena.com (http://www.gsmarena.com/nokia-phones-1.php)

```
      publicclass checkboxselect {

      publicstaticvoid main(String[] args) {
            WebDriver driver = new FirefoxDriver();
            driver.get("http://www.gsmarena.com/nokia-phones-1.php") ;
            List<WebElement>chbox
=driver.findElements(By.xpath("//input[@type='Checkbox']")) ;
            //Select the check box
            for(int i=0 ; i <chbox.size(); i++){
                  chbox.get(i).click();
            }
            //Deselect the checbox using foreach
            for(WebElement chbox1:chbox){
                  chbox1.click();
            }
            driver.close();
      }

}
```

4) Write a script to select all the alternative checkbox present on the webpage GSMarena.com

```
      publicclass alternativechboxselect {

      publicstaticvoid main(String[] args) {
            WebDriver driver = new FirefoxDriver() ;
            driver.get("http://www.gsmarena.com/nokia-phones-1.php");
            List<WebElement> chboxalt
=driver.findElements(By.xpath("//input[@type='Checkbox']"));
```

```
            for(int i =0 ;i < chboxalt.size();i =i+2){
                  chboxalt.get(i).click();
            }

            driver.close();
      }

}
```

5) Write a script to select all the checkbox present in the webpage in reverse order.

```
      publicclass reveresechkselect {

      publicstaticvoid main(String[] args) {
            WebDriver driver = new FirefoxDriver();
            driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS) ;
            driver.get("http://www.gsmarena.com/nokia-phones-1.php") ;
            List<WebElement> chboxreverse
=driver.findElements(By.xpath("//input[@type='Checkbox']")) ;
            for(int i =chboxreverse.size()-1 ; i>=0 ;i--){
                  chboxreverse.get(i).click();
            }

     driver.close();
      }

}
```

6) Write a script to select first and last checkbox present on the webpage?

```
      publicclass firstlastchbox {

      publicstaticvoid main(String[] args) {
            WebDriver driver = new FirefoxDriver();
            driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
            driver.get("http://www.gsmarena.com/nokia-phones-1.php") ;
            List<WebElement> firstlast
=driver.findElements(By.xpath("//input[@type='Checkbox']"));

            firstlast.get(0).click();
            firstlast.get(firstlast.size()-1).click() ;



      }

}
```

7) Write a script to count no of images present in the webpage.

```
publicclass countimages {

      publicstaticvoid main(String[] args) {
            WebDriver driver = new FirefoxDriver() ;
```

```
		driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS) ;
		driver.get("http://www.gsmarena.com/nokia-phones-1.php") ;
		List<WebElement> coutimgs = driver.findElements(By.xpath("//img")) ;
		for(int i =0; i <coutimgs.size(); i++) {
			String imges = coutimgs.get(i).getAttribute("title");
			System.out.println(imges);
		}

		int count =coutimgs.size() ;
		System.out.println(count);

		driver.close();
	}

	}
```

Handling the listbox: in webelement interface we do not have any appropriate method, to handle the list box, in order to perform required action on list box we use **Select** Class

Frist find the list boxElement ausing findElement() method, then pass it an argument for Select class constructor and call any one of the select By method of the select class as shown in the below.

```
	publicclass Selectlistbox {

	publicstaticvoid main(String[] args) {
		WebDriver driver = new FirefoxDriver() ;
		driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
		driver.get("http://www.fatcow.com/");
		 WebElement elemt = driver.findElement(By.id("countrySelect"));
		 Select select =new Select(elemt);
		 select.selectByVisibleText("Australia");
		 driver.close();



	}

	}
```

We can use same selectBymethod to handle Multi select list box .

1) Write a script to count the number of option present in the list box.

```
	publicclass countlist {

	publicstaticvoid main(String[] args) {
		WebDriver driver = new FirefoxDriver() ;
		driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
		driver.get("file:///C:/selenium/HTMLfiles/ListBoxmypage.html") ;
		WebElement listbox = driver.findElement(By.id("countrySelect")) ;
		Select cslect =new Select(listbox) ;
		int count = cslect.getOptions().size() ;
		System.out.println(count);
		driver.close();

	}

}
```

2) Write a script to print all the option present in the list box (fatcow.com)

```
publicclass listoptionprint {

publicstaticvoid main(String[] args) {
	WebDriver driver = new FirefoxDriver() ;
	driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
	driver.get("file:///C:/selenium/HTMLfiles/ListBoxmypage.html");
	WebElement listbox = driver.findElement(By.id("countrySelect"));
	Select oselect = new Select(listbox);
	int countlist = oselect.getOptions().size() ;
	for(int i=0; i <countlist ; i++) {
		System.out.println(oselect.getOptions().get(i).getText());

	}
		driver.close() ;

}
```

3) Write a script to print all the selected options with multi selected list box in reverse order.

```
publicclass multiselectedreverese {

publicstaticvoid main(String[] args) {
	WebDriver driver = new FirefoxDriver();
	driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
	driver.get("file:///C:/selenium/HTMLfiles/ListBoxmypage.html");
	WebElement listbox = driver.findElement(By.id("countrySelect"));
	Select aselect =new Select(listbox);
	for(int i =aselect.getOptions().size()-1; i >=0 ;  i--)
	{
		System.out.println(aselect.getOptions().get(i).getText());
	}

driver.close();

}
```

4) Write a script to print name and text of the first selected option present in the multiselected list box

```
publicstaticvoid main(String[] args) {
	WebDriver driver = new FirefoxDriver() ;
	driver.manage().timeouts().implicitlyWait(10,TimeUnit.DAYS) ;
	driver.get("file:///C:/selenium/HTMLfiles/ListBoxmypage.html");
```

```
		WebElement listbox = driver.findElement(By.id("countrySelect"));
		Select aselect =new Select(listbox) ;
		intcountsize =aselect.getOptions().size() ;
		List<WebElement> selectoption = aselect.getAllSelectedOptions();
		for(int i=0; i< selectoption.size(); i++){
			System.out.println(selectoption.get(i).getText());

		}

		if(aselect.isMultiple()){
			System.out.println("yes multiple list box");
		}
			else{
				System.out.println("not a mulit select list box ");
			}
		}

	}
```

3) Write a script verify whether the given list box is mulityselect list box dropdown combo box.

```
	publicclass multyorcombo {

	publicstaticvoid main(String[] args) {

		WebDriver driver = new FirefoxDriver();
		driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
		driver.get("http:\\www.fatcow.com");
		WebElement listbox = driver.findElement(By.id("countrySelect"));
		Select listselect = new Select(listbox);
		if(listselect.isMultiple()){
			System.out.println("is smultilistbox");
		}
		else{
			System.out.println("not multiselected");
		}

		driver.close();

	}

}
```

4) Write a script to select all the option present in the mulitselect list box.

```
	publicclass Selectedoptiosn {


	publicstaticvoid main(String[] args) {
		WebDriver driver = new FirefoxDriver() ;
		driver.manage().timeouts().implicitlyWait(10,TimeUnit.DAYS) ;
		driver.get("file:///C:/selenium/HTMLfiles/ListBoxmypage.html");
		WebElement listbox = driver.findElement(By.id("countrySelect"));
```

```
            Select aselect =new Select(listbox) ;
            intcountsize =aselect.getOptions().size() ;
            List<WebElement> selectoption = aselect.getAllSelectedOptions();
            for(int i=0; i< selectoption.size(); i++){
                  System.out.println(selectoption.get(i).getText());

            }

            if(aselect.isMultiple()){
                  System.out.println("yes multiple list box");
            }
                  else{
                        System.out.println("not a mulit select list box ");
                  }
            driver.close();
            }


      }
```

Note: in Select class we gave deselect() methods, which is simllar to selectByMethods() such as
 deSelectAll();
deSeletByValue() ;
deSelectByIndex();
deSelectByvisibleText() ;

But there is no select All method in order to Select all the option we should write a looping code as shown in the above example.

5) Write a script to search for the Specified option present in the List box.

```
      publicclass specifiedoption {

      publicstaticvoid main(String[] args) {
            WebDriver driver = new FirefoxDriver();
            driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
            driver.get("file:///C:/selenium/HTMLfiles/ListBoxmypage.html");
            WebElement listbox = driver.findElement(By.id("countrySelect"));
            String eval ="Saab" ;
            String msg ="not found" ;
            Select selectmsg = new Select(listbox);
            List<WebElement> listitem = selectmsg.getOptions() ;
            for(int i =0 ; i< listitem.size() ; i++) {
                  String avalue =listitem.get(i).getText() ;
                  if(eval.equalsIgnoreCase(avalue)){
                        msg ="Found @index :"+ i ;
                        break ;
                  }
            }
            System.out.println(msg);
```

```
driver.close();
}

}
```

1) What is dynamic list box how to handle it , list box whoes content is keep changing during runtime is called as dynamic list box, in order to handle them use Select class itself. Along with Explicit wait.

   Note: Most of the class dynamic element or developed using AJAX (Asynchrons java script and XML) technology to handle Ajax element we use Explicit wait().

   Ex: Selecting the required in country list box which in state then selecting respective state, in the state list box it is dynamic .

```
    publicclass dyanmiclistbox {


publicstaticvoid main(String[] args) {
        WebDriver driver = new FirefoxDriver() ;
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        driver.get("http://www.plus2net.com/php_tutorial/ajax-dd3.php");
     WebElement listboxcountry = driver.findElement(By.id("s1")) ;
        Select clist = new Select(listboxcountry) ;
        clist.selectByVisibleText("IND") ;
        for(int i =0; i< clist.getOptions().size(); i++)
        {
                System.out.println(clist.getOptions().get(i).getText());
        }

        WebElement stlistbox = driver.findElement(By.name("state"));
        Select slist = new Select(stlistbox) ;
        System.out.println(slist.getOptions().size());
        for(int i =0 ;i < slist.getOptions().size(); i++){

System.out.println(slist.getOptions().get(i).getAttribute("value"));
        }

        WebDriverWait wait = new WebDriverWait(driver, 15);
        By b = By.xpath("//option[@value='Gujarat']");
        wait.until(ExpectedConditions.elementToBeClickable(b));
        slist.selectByVisibleText("Madhya Pradesh") ;

        //By.xpath("//option[@value='Gujarat']") ;

        driver.close();


}

    }
```

Handling customized List box any list box which is developed without using Select HTML tag is called as customized list box.

We can’t handle the customized list box, using Select class we try to use it, we get unexpected tag name Exception.

In order to handle it we use Send key’s method itself as shown below.

```
publicclass custimzedlistbox {

	publicstaticvoid main(String[] args) {
		WebDriver driver = new FirefoxDriver() ;
		driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
		driver.get("http://www.yatra.com/");
		WebElement listbox1 = 
driver.findElement(By.id("BE_flight_origin_city"));
		listbox1.sendKeys("Bangalore");
		listbox1.sendKeys(Keys.ENTER);

		WebElement arlist = driver.findElement(By.id("BE_flight_arrival_city"))
;
		arlist.sendKeys("Hyderabad") ;
		arlist.sendKeys(Keys.ENTER);


	}

    }
```

2) Write a script to search selenium in google and print all the autosuggested options.

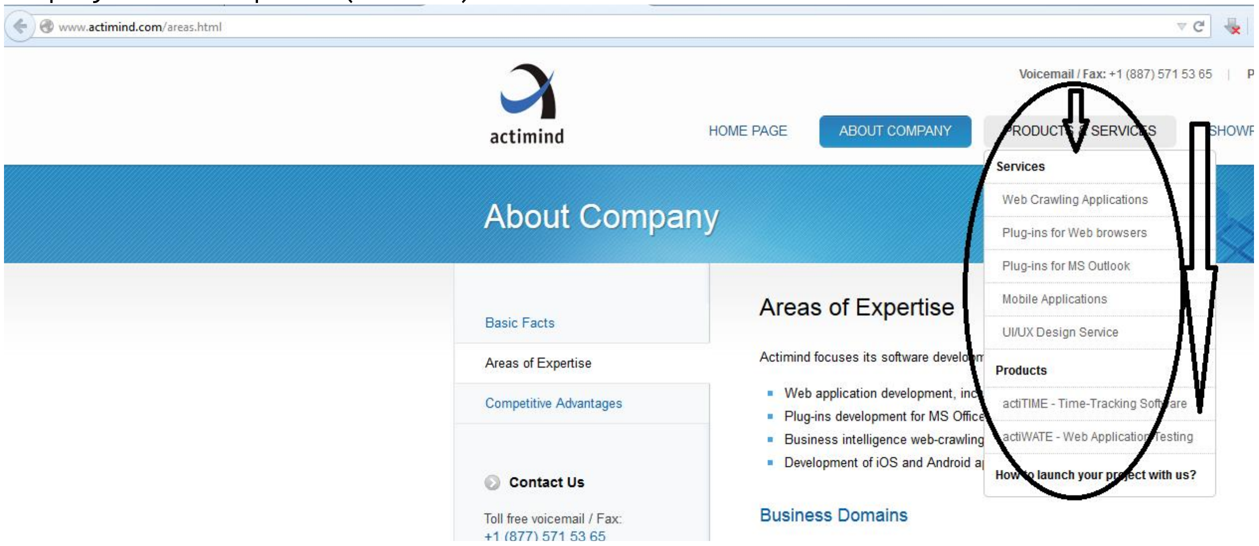
```
	publicclass searchprintautolist {

	publicstaticvoid main(String[] args) {
		WebDriver driver = new FirefoxDriver() ;
		driver.manage().timeouts().implicitlyWait(20,TimeUnit.SECONDS) ;
		driver.get("http:\\www.google.com") ;
		WebElement intextbox = driver.findElement(By.id("gbqfq"));
		intextbox.sendKeys("Selenium") ;

		String xp = "//div[@class='sbqs_c']" ;
		List<WebElement> autolist = driver.findElements(By.xpath(xp));
		for(int i =0 ; i< autolist.size(); i++){
			//System.out.println(autolist.get(i).getAttribute("class"));
			System.out.println(autolist.get(i).getText()) ;
		}

		driver.close();
	}

    }
```

3) Write a script to search selenium in the google and select the autosuggestion if it contains the tutorial then click or select.

```
	publicclass chkautolist {

	publicstaticvoid main(String[] args) {
		WebDriver driver = new FirefoxDriver() ;
		driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
		driver.get("http:\\www.google.com") ;
		WebElement inputbox = driver.findElement(By.xpath("//input[@id
='gbqfq']"));
		inputbox.sendKeys("Selenium");
		List<WebElement> autolist = 
driver.findElements(By.xpath("//div[@class='sbqs_c']"));
		for(int i =0; i < autolist.size(); i++){
			String actal =autolist.get(i).getText();
			System.out.println(actal);
			if (actal.contains("tutorial")){
				autolist.get(i).click();
				break ;
			}
		}
	}

}
```

Note: if we try to perform any action on the element after the Element is reloaded or disappeared or page refresh then we get **StaleElementRefrence Exception**

Drop down Menu: it is an element on which of we move the mouse pointer it will display list of option (submenu).


www.actimind.com/areas.html
Voicemail / Fax: +1 (887) 571 53 65
actimind
HOME PAGE
ABOUT COMPANY
PRODUCTS & SERVICES
SHOWR
About Company
Services
Web Crawling Applications
Plug-ins for Web browsers
Plug-ins for MS Outlook
Mobile Applications
UI/UX Design Service
Products
actiTIME - Time-Tracking Software
actiWATE - Web Application Testing
How to launch your project with us?
Basic Facts
Areas of Expertise
Competitive Advantages
Contact Us
Toll free voicemail / Fax:
+1 (877) 571 53 65
Areas of Expertise
Actimind focuses its software developm
Web application development, inc
Plug-ins development for MS Office
Business intelligence web-crawling
Development of iOS and Android a
Business Domains

In order to handle drop down menu we use **Actions**() class. In order to move the mouse pointer on the element we use. Move to element method of Action class But whenever we call any action class it is mandatory, that we call perform().method

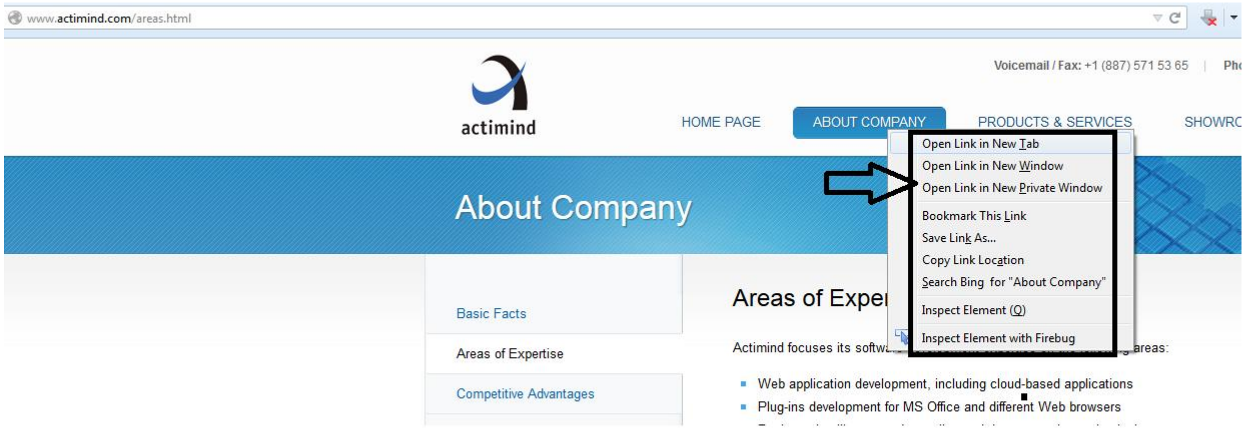
Ex:

1) Select “basic back” option menu in present in the www.actimind.com (http://www.actimind.com/basic-facts.html)

```
publicclass Dropdownmenuaction {

	publicstaticvoid main(String[] args) {
		WebDriver driver = new FirefoxDriver() ;
		driver.manage().timeouts().implicitlyWait(20,TimeUnit.SECONDS);
		driver.get("http://www.actimind.com/basic-facts.html");
		WebElement menu = driver.findElement(By.xpath("//span[text()='About 
Company']"));
		Actions actions = new Actions(driver);
		actions.moveToElement(menu).perform() ;
		
		//driver.findElement(By.xpath("//a[text()='Basic Facts']")).click();
		driver.findElement(By.xpath("//a[text() ='Areas of 
Expertise']")).click() ;
		driver.close();
		
		
		
		
	}

}
```

Context Menu: The list of option displayed when you right click on the page is called as context menu:

In order to right click on the webpage, we use context click method of Action class(contextclick) But **selenium cannot recognize option present in the context menu.**


www.actimind.com/areas.html
Voicemail / Fax: +1 (887) 571 53 65
actimind
HOME PAGE
ABOUT COMPANY
PRODUCTS & SERVICES
SHOWRC
Open Link in New Tab
Open Link in New Window
Open Link in New Private Window
Bookmark This Link
Save Link As...
Copy Link Location
Search Bing for "About Company"
Inspect Element (Q)
Inspect Element with Firebug
About Company
Basic Facts
Areas of Expertise
Competitive Advantages
Areas of Exper
Actimind focuses its softw
areas:
Web application development, including cloud-based applications
Plug-ins development for MS Office and different Web browsers

In order to select the required option present in the context menu. Type the short cut keys “such as “T” for new Tab, “w” for new Window, etc.. using sendkeys() methods of Actions class as shown below.

```
publicclass contextmenu {

	publicstaticvoid main(String[] args) {
```

```
            WebDriver driver = new FirefoxDriver() ;
            driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
            driver.get("http://www.actimind.com/basic-facts.html");
            WebElement homepage = driver.findElement(By.xpath("//span[text() ='Home
Page']"));
            Actions actionmenu = new Actions(driver);
            actionmenu.contextClick(homepage).perform() ;
            actionmenu.sendKeys("w").perform() ;

            driver.close();

      }

}
```

Performing Drag and drop in order to another location we use drag and drop method of the action class.

```
            actionmenu.dragAndDrop(source, target);
```

2) Write a script to drag1 to block 1 and drop it on to block3 which is presenting .

http://www.dhtmlgoodies.com/submitted-scripts/i-google-like-drag-drop/index.html

```
   publicclass dragdrop {

   publicstaticvoid main(String[] args) {
         WebDriver driver = new FirefoxDriver() ;
         driver.manage().timeouts().implicitlyWait(20,TimeUnit.SECONDS);
               driver.get("http://www.dhtmlgoodies.com/submitted-scripts/i-
   google-like-drag-drop/index.html");
         WebElement block1 = driver.findElement(By.xpath("//h1[text()='Block
   1']"));

               WebElement block3 = driver.findElement(By.xpath("//h1[text()
   ='Block 3']"));
         Actions act =new Actions(driver);
         act.dragAndDrop(block1, block3).perform();

         driver.close();


   }

}
```

Note: Actions class can also be used double.click statement.
Actions.doubleclick(webelemnet).perform()

Handling Pop-up:

In selenium performing action on the pop-up depended on the type of the pop-up. Some of the pop-up cannot be handled by selenium in such cases we use alternative option and 3rd party tools.

In order to Clearyidentifypop-up type always use the browser. It is opened by selenium, we can categorize the pop-up window following types.
1) Alert confirmation
2) Hidden Division pop-up
3) Page on-load pop-up
4) File upload pop-up
5) File download pop-up
6) Child browser pop-up
7) Window pop-up

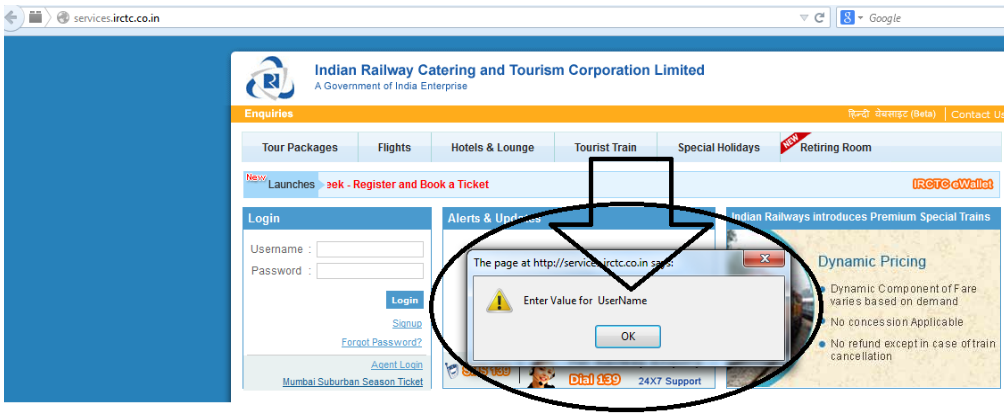
1) Write a script take screen shot of the application .png format

```
publicclass captureimage {

   publicstaticvoid main(String[] args) throws IOException {
          WebDriver driver = new FirefoxDriver();
          driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
          driver.get("http:\\www.google.com");

           EventFiringWebDriver edriver = new EventFiringWebDriver(driver);
           File srcFile = edriver.getScreenshotAs(OutputType.FILE);

           FileUtils.copyFile(srcFile, new File("C://test/google.png"));
           driver.close();


   }

}
```

Note:

1. The getScreenShotAs() method takes complete Screenshot of the webpage(it will automatically does the scrolling .
2. In selenium there is no option to take the screen shot of specific element.
3. We cannot take the screen shot of the only logo present on the webpage.

Alert and confirmation pop-up:


services.irctc.co.in
Indian Railway Catering and Tourism Corporation Limited
A Government of India Enterprise
Enquiries
Tour Packages
Flights
Hotels & Lounge
Tourist Train
Special Holidays
Retiring Room
Login
Username :
Password :
Login
Signup
Forgot Password?
Agent Login
Mumbai Suburban Season Ticket
Indian Railways introduces Premium Special Trains
Dynamic Pricing
The page at http://services.irctc.co.in says:
Enter Value for UserName
OK

Characteristics:
1) We can move the pop-up
2) We cannot inspect the pop-up
3) It will have ok button
4) If the ok and cancel button this is confirmation pop-up.

In order to handle the pop-up first we should transfer control to the Alert pop-up using switch start it returns the object of type alert and we can use following method of Alert class.
1) GetText() used to get the text of the pop-up.
2) Accept – used to click on "ok' button
3) Dismiss- > used to click on cancel button

Note:
1) If Alert pop-up is not present it will throw **NoAlretPresentException:**
2) After closing the alert pop-up control will be transferred back to the mainpage automatically.
3) There cannot be multiple alert pop-up at the same time, that is if multiple pop –up displayed at the same time it will be other than the Alert pop-up.
4) Most of the cases will be hidden division pop-up or child browser pop-up.
5) If multiple alert pop-up displayed, then each time we should use switch to statement.

Ex:
```
publicclass alertconfimation {


      publicstaticvoid main(String[] args) {

            WebDriver driver = new FirefoxDriver() ;
            driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
            driver.get("http://services.irctc.co.in/");

            //driver.findElement(By.id("button")).click();
            //driver.findElement(By.name("button")).click() ;
            //driver.findElement(By.cssSelector("input[id ='button']")).click() ;
            driver.findElement(By.xpath("//input[@id ='button']")).click() ;

            try{
                    Alert alret = driver.switchTo().alert();
                    System.out.println("Alert is present");
                    String msg = alret.getText() ;
                    System.out.println(msg);
                    alret.accept() ;


             }
             catch(NoAlertPresentException e ){
                    System.out.println("Alret is not present");
             }
             driver.close();
      }

}
```

Note: Java script pop-up is Alert pop –up

Hidden division pop-up:
Characteristics:

1) Most of the case we cannot move the pop-up
2) We can inspect the pop-up
3) It can have any type of buttons or buttons may not be present
4) Most of the case’s it will colorful

Since we are allowed to inspect the element we can handle this pop-up using findElement() method itself.

Note: This is called hidden division pop-up because it is created using Html tag div and initially its hidden (Display none)

2) Calendar pop-up is type of hidden division pop-up

Ex:

```
publicclass hiddendivisionpopup {


      publicstaticvoid main(String[] args) {
      WebDriver driver = new FirefoxDriver() ;
      driver.manage().timeouts().implicitlyWait(20,TimeUnit.SECONDS) ;
      driver.get("http://www.2shared.com/") ;
      //driver.findElement(By.cssSelector("input[type='image']")).click() ;
      driver.findElement(By.xpath("//input[@title ='Upload file']")).click() ;


      /*input id="login" type="text" size="22" name="login"
                 Any of the below find element can be used.
                    By.id("login");
                    By.name("login");
                    By.cssSelector("input[id = 'login']")
                    xpath = //input[@id ='login'] */

      try {
             driver.findElement(By.id("login")).sendKeys("vinay2ml@gmail.com") ;
             driver.findElement(By.id("password")).sendKeys("nokianokia123") ;
             driver.findElement(By.id("password2")).sendKeys("nokianokia123") ;
```

```
		driver.findElement(By.xpath("//button[@onclick='return 
doSignUp();']")).click();
		//id="loginErrorMsg" class="body" align="left" style="color:red" 
colspan="2"
		String msg = 
driver.findElement(By.id("loginErrorMsg")).getAttribute("id");

		if(msg.equalsIgnoreCase("loginErrorMsg")){
			driver.findElement(By.linkText("I already have account")).click()
;
		}

		driver.findElement(By.id("login")).sendKeys("vinay2ml@gmail.com");
		driver.findElement(By.id("password")).sendKeys("nokianokia123") ;
		driver.findElement(By.xpath("//button[@onclick='return 
doLogIn();']")).click() ;


	}

	catch(NoSuchElementException e){
		System.out.println("popup is not displayed");
	}
	}

}
```

1) How do you handle the calendar pop-up ?
   It is hidden division pop-up we handle it findElement() method itself.

```
	publicclass calenderhandle {


	publicstaticvoid main(String[] args) {
		WebDriver driver = new FirefoxDriver() ;
		driver.manage().timeouts().implicitlyWait(20,TimeUnit.SECONDS) ;
		driver.get("http://www.yatra.com/");

	driver.findElement(By.xpath("//input[@id='BE_flight_depart_date']")).click();
		driver.findElement(By.xpath("//a[@id='a_2014_10_15']")).click() ;



	}

	}
```

2) How do you enter into the Textbox without using SendKeys() using java script.

Using JavaScript we can enter the text in the text box even if the textbox is disabled.

3) Write a script to select future date in the calendar without writing the navigation code.

4) Write a script to select today's date in the calendar presetting in yatra.com
5)

Page on load pop-up:

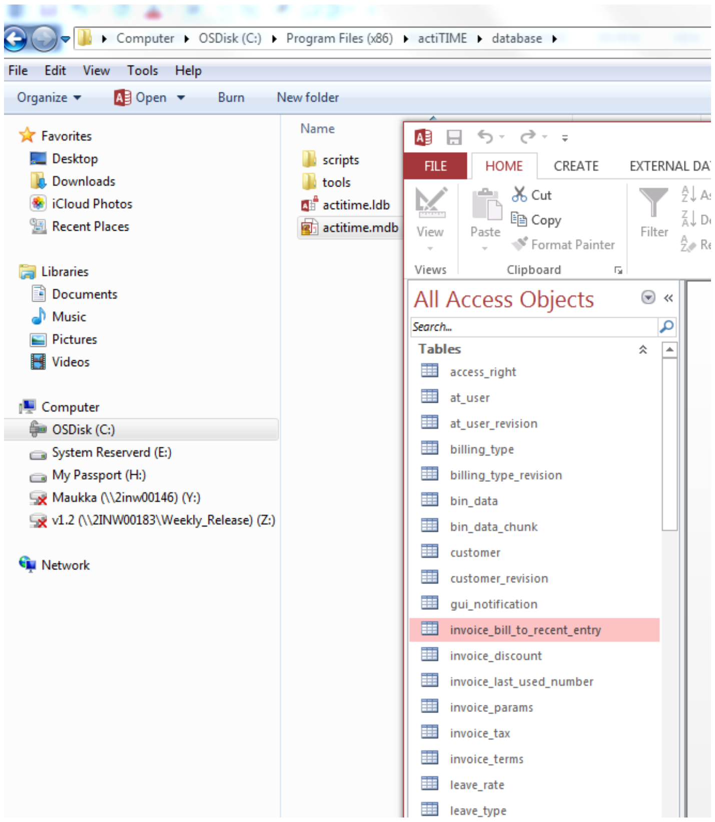
1) We can move the pop-up
2) We cannotinspect the pop-up
3) It will have "ok" and "cancel" button
4) It will have username and password filed.

Note: this is pop-up is displayed while loading the webpage, hence it is called as page onload pop-up.

Solution: Selenium can 'not perform any action on page on-load pop-up in order to handle it the specify user name and password in the url itself as shown in below.

Handling Database:
While testing the application some time we need to verify the data present in the database also. In order to do this we should have the following information of the database such as :
1) Data base type
2) Data base location
3) Username and password
4) Login to data base
5) Table structure (Data schema) and SQL knowledge

Ex: Actitime Application:
1) Data base type = MS Access
2) Data base location = "C:\Program Files (x86)\actiTIME\database\"
3) Username and password : no username and pass
4) Table structure : ex: access_right, at_user, customer etc..


Computer ▸ OSDisk (C:) ▸ Program Files (x86) ▸ actiTIME ▸ database ▸
File Edit View Tools Help
Organize Open Burn New folder
Favorites
Desktop
Downloads
iCloud Photos
Recent Places
Libraries
Documents
Music
Pictures
Videos
Computer
OSDisk (C:)
System Reserverd (E:)
My Passport (H:)
Maukka (\\2inw00146) (Y:)
v1.2 (\\2INW00183\Weekly_Release) (Z:)
Network
Name
scripts
tools
actitime.ldb
actitime.mdb
FILE HOME CREATE EXTERNAL DA
View Paste Cut Copy Format Painter Filter
Views Clipboard
All Access Objects
Search...
Tables

access_right
at_user
at_user_revision
billing_type
billing_type_revision
bin_data
bin_data_chunk
customer
customer_revision
gui_notification
invoice_bill_to_recent_entry
invoice_discount
invoice_last_used_number
invoice_params
invoice_tax
invoice_terms
leave_rate
leave_type

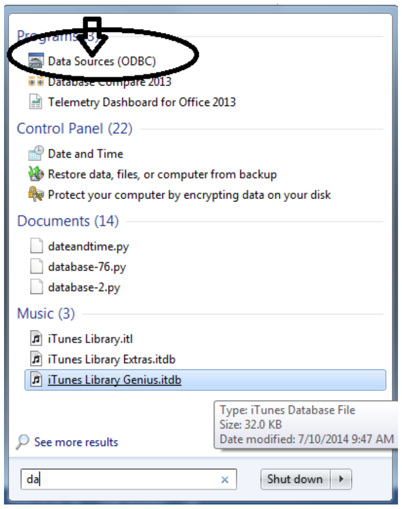
5) Ex: sql statement :"Select * from access_right"

While retrieving the information from database programaitcally, we should provide information such as database type, location , username and password . which are called as **connection string :**

The connection string can be stored in a system variable called DSN (Data source name)

In order to see the DSN available in the system
Start→Data base connection

Vinay2ml@gmail.com

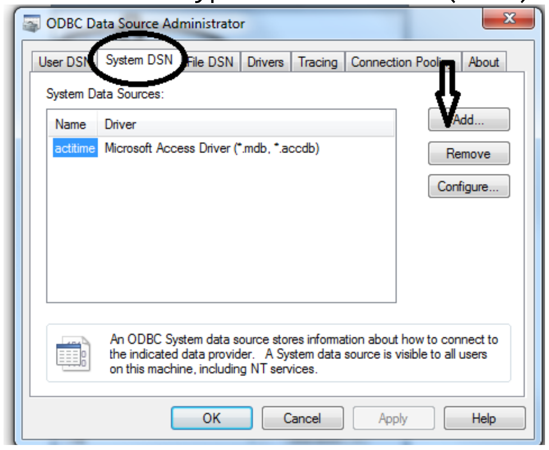


Type: iTunes Database File
Size: 32.0 KB
Date modified: 7/10/2014 9:47 AM

We can create our own DSN connect to any required data base before creating the DSN assume that required database driver installed in the system .

Steps to create a DSN:

1) Go to start type: data source (ODBC) and select DSN



System Data Sources:

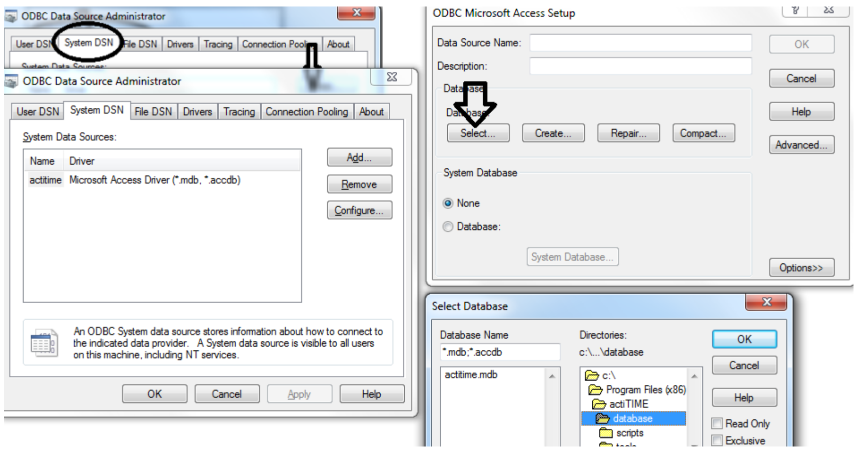
Name	Driver
actitime	Microsoft Access Driver (*.mdb, *.accdb)

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

2) Specify the data source name :

3)

Vinay2ml@gmail.com



Name	Driver
actitime	Microsoft Access Driver (*.mdb, *.accdb)

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

Database Name: *.mdb;*.accdb — actitime.mdb

Directories: c:\...\database — c:\, Program Files (x86), actiTIME, database, scripts

Note: in order to specify the username and password click on Advanced button.

1) Java code to open and close database connection.

```
publicclass connetingdatabase {

	publicstaticvoid main(String[] args) throws SQLException {

		Connection con = DriverManager.getConnection("jdbc:odbc:actitime") ;
		con.close() ;
		System.out.println("connection successfuly done");

	}

}
```

2) Write a code to count the no of column present in the table and also print name of the column

```
	publicclass countcolumnname {
	publicstaticvoid main(String[] args) throws SQLException {
		Connection c = DriverManager.getConnection("jdbc:odbc:actitime");
		String sql = "select * from customer" ;
		ResultSet r = c.createStatement().executeQuery(sql);
		int cc = r.getMetaData().getColumnCount() ;
		System.out.println(cc);

		for (int i =1 ;i <= cc ; i++) {
			String cname = r.getMetaData().getColumnName(i) ;
			System.out.println(cname);
		}

		c.close() ;

	}

}
```

3) Write a code to print the content of the specified table

```
publicclass specifiedtable {

    publicstaticvoid main(String[] args) throws SQLException {
      Connection con = DriverManager.getConnection("jdbc:odbc:actitime") ;
      String sql = "Select * from customer" ;
      ResultSet row = con.createStatement().executeQuery(sql) ;
    int cc = row.getMetaData().getColumnCount();
    while(row.next()){

           for(int i = 1; i<=cc ; i++)
             {
                   String data = row.getString(i) ;
                   System.out.println(data);
             }
      }



      con.close();



    }

}
```

Note:

1) The above program can be used on any database on any system only DSN name changes.
2) In JDBC column index start from 1 like xpath.
3) ResultSet do not have the hasnext() method ,next() method itself perform the action of the hasnext() method
4) To retrieve data present in the cell of the table we alwasys use getString() method irrespective of the cell data type. Later we may use wrapper class or type casting to convert the data into required type.

TestNG : (Next Generation)

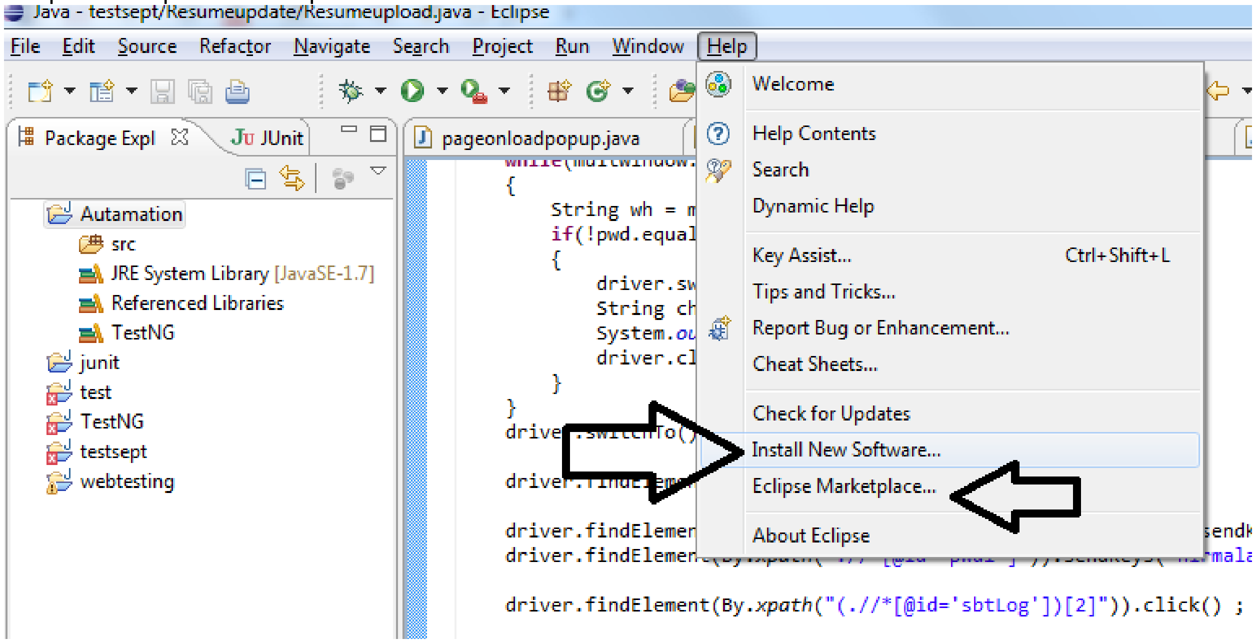
TestNG is a unit testing tool. Basically used by developer to perform white box testing that is every requirement given by the customer , developer develops a java class which will be called as business class. If there are multiple requirements a there will be multiple business class.

In order to test all the business class (white box testing) developer develops corresponding test class in order to run all the test class together(Batch execution) developer uses TestNG. Which will also generate execution result automatically in html format it also as other built in functions , such as : Assertion , Re-execution of the failure scripts etc..

In selenium for every manual test cases we develop automation script in java class. In order to execute all the selenium scripts to generate execution results etc. we also use TestNg.

Vinay2ml@gmail.com

Installing TestNG :
1) Open Eclipse go to help
2) Help→ Eclipse Marketplace →Search → Test NG



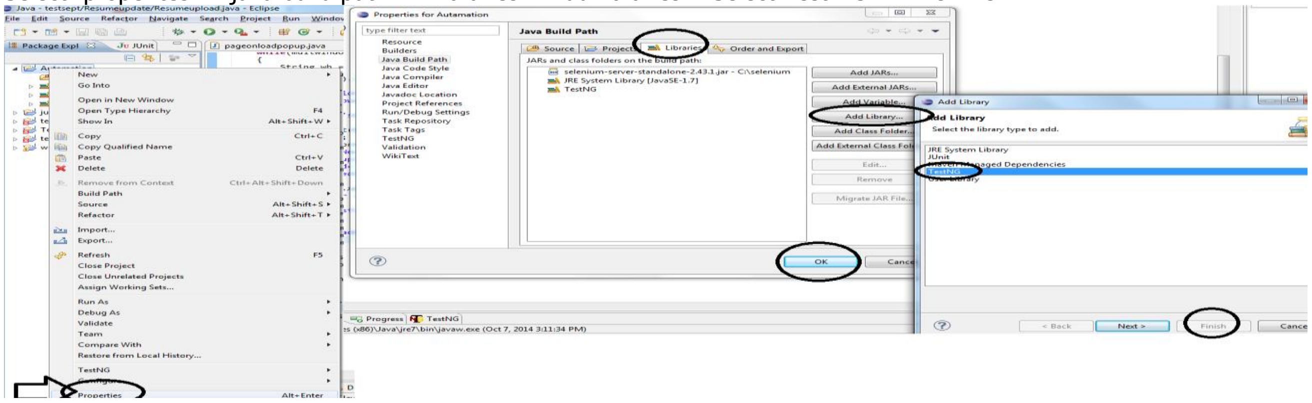
```
{
    String wh = m
    if(!pwd.equal
    {
        driver.sw
        String ch
        System.ou
        driver.cl
    }
}
driver.switchTo()

driver.findElem

driver.findElemen
driver.findElement(By.xpath(
driver.findElement(By.xpath("(.//*[@id='sbtLog'])[2]")).click() ;
```

3) Click install button of Test NG for Eclipse
4) Accept->Finish -> ok for warning message
5) Yes -on the confirmation.

Note: if Eclipse marketplace is not available then select “install New software” under the help menu . click→Add→ specify the name as TestNG location as : http://beust.com/eclipse. Get the location url from http://testng.org/doc/download.html web site.

Configuring the java project:
1) Right click on the java project
2) Select- properites -> java build path -> Libraries →Add libraries →Select Test NG →Finish—OK.

3)

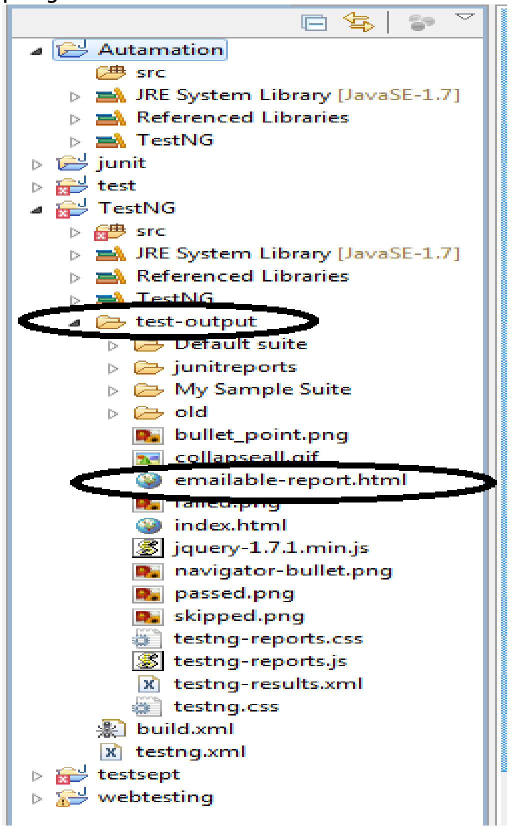


Test NG class:

1) It is a java class which contains test Method .

2) Any method which is written below Test Annotation is called as Test Method.
3) @Test is called as Test Annotation syntactic metadata

Whenever we run any TestNG class it automatically generate the result in Html format (Emailable-reprot.html) or index html. Inside (Test-output) folder of the java project



```
Autamation
    src
    JRE System Library [JavaSE-1.7]
    Referenced Libraries
    TestNG
junit
test
TestNG
    src
    JRE System Library [JavaSE-1.7]
    Referenced Libraries
    TestNG
    test-output
        Default suite
        junitreports
        My Sample Suite
        old
        bullet_point.png
        collapseall.gif
        emailable-report.html
        failed.png
        index.html
        jquery-1.7.1.min.js
        navigator-bullet.png
        passed.png
        skipped.png
        testng-reports.css
        testng-reports.js
        testng-results.xml
        testng.css
    build.xml
    testng.xml
testsept
webtesting
```

If the test-output folder is not present it will create it , if already present it will be overwrites . in order to write any information into html report as wee as console use log method of Reporter.class as shown in the below.

```
package com.qspider;

import org.testng.ITestResult;
import org.testng.Reporter;
import org.testng.annotations.Test;

publicclass demo {



	privatestaticfinal ITestResult True = null;

	@Test
	publicvoid testA()
	{
		Reporter.log("Login");
	}

}
```

Note: in TestNG main method is present in org.testing.TestNG .

1) Can we have multiple Test Method in the TestNG class ?
   Ans: Yes

2) If there are multiple Test method what is the order of execution ?
   Ans: Alphabetical order

3) Then how do you execute Test method which is required order ?
   Using Priority

```
package com.qspider;

import org.testng.Reporter;
import org.testng.annotations.Test;

publicclass priorityexp {

	@Test(priority = 2)
	publicvoid Deletecustomer()
	{
		Reporter.log("Delete the custoer") ;

	}

	@Test( priority = 3)
	publicvoid Editcustomer()
	{
		Reporter.log("Edit the custoer ");

	}
	@Test(priority = 1)
	publicvoid Registercustomer()
	{
		Reporter.log("Create the customer ");
	}

}
```

Note:

1) giving the priority is not mandatory, if it is not provided its priority will be zero (0)
2) it can give the negative number also for priority
3) Then priority is need not be in sequential order
4) The priority should always a number , it always execute in ascending order.

1) How do you create dependency in the TestNG
   Ans: dependsonMethods

Ex:

```
package com.qspider;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.Test;

publicclass dependsonmethodex {

	@Test(dependsOnMethods ="Registercustor")
	publicvoid deletecustoer()
	{
		Reporter.log("Delete custer depend on register") ;
	}

	@Test
	publicvoid Registercustor()
	{
		Reporter.log("Register custoremr ") ;
		Assert.assertTrue(false) ;

	}
}
```

Note: in the above ex: frist it wil try to execute delete customer (because of alphabetical order), but because of dependency it will execute first registercustomer method. Which will fail during the runtime , hence it will skip execution of delete customer method.

output:

```
[TestNG] Running:
  C:\Users\viml\AppData\Local\Temp\testng-eclipse-142572682\testng-customsuite.xml

FAILED: Registercustor

SKIPPED: deletecustoer

===============================================
    Default test
    Tests run: 2, Failures: 1, Skips: 1
```

2) When the TestNG class is skip -> if the dependsonMethod is failed it will skip the test method.

3) How to create multiple dependency or how to make a method depend on multiple methods ?
   Ans: (dependsOnMethods ={“Registercuster”,”Editcostmer”})

1) How do you perform validation in TestNG ?

Using Assert class validation refers to comparing actual value with expected value and reporting the status.

Ex:

```
package com.qspider;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.Test;

publicclass assertequalscheck {

	@Test
	publicvoid Registercustormer(){

		Reporter.log("Login") ;
		Reporter.log("Register custmer") ;
		String msg = "Created customer" ;
		String emsg ="Created not customer" ;
		Assert.assertEquals(msg, emsg);
		Reporter.log("Logout");
	}
}
```

Note: when we use Assertion it will compare the actual value with expected value of both are same then the status will be pass, and it will continue the execution of the reaming statement.
	If actual and expected values are different the status will be failed and it will stop the current test method execution. (But it will continue execution reaming methods).

Important methods of Assert class and static methods.
1) assertEquals()
2) assertNotEquals()
3) assertTrue()
4) assertFalse()
5) assertNull()

1) How do you ensure that Test will continue execution ever after the verification failed.?
Using SoftAssert().

Ex:

```
publicclass usingsoftassert {

	@Test
	publicvoid registercustormer()
	{
	SoftAssert softassert = new SoftAssert() ;
	Reporter.log("Login") ;
	Reporter.log("register");
	String msg ="created customer" ;
	String emsg ="created" ;
	softassert.assertEquals(msg, emsg);
```

```
        Reporter.log("logout") ;
        softassert.assertAll();

        }

}
```

2) What is the difference between Assert and softAssert

Assert	softAssert
1) It will stop the current method execution when the verification failed	1) It will not stop the current method execution when the verification failed
2) All the method are static ,we don't create the object	2) All the methods are non-static, we have to create the object
3) We do not call assertAll() method	3)we must call assertAll() method

Note: while verifying the important or critical features use assert that is the steps where can not procced, because of the failure like blocker defect or showstopper. But otherwise we can use softassert() method.

Important Annotiaon of TestNG:

1) @Test : this indicate the Test method
2) @BeforeMethod : This method will be executed before the execution of every test method
3) @AfterMethod : This method will be executed after the execution of every test method.
4) BeforeClass : This method will be executed only once beginning of the class
5) AfterClass : This method will be executed only once at the end of the class.

Ex:

```
package com.qspider;

import org.testng.Reporter;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

publicclass DemoA {
        @BeforeClass
        publicvoid openApp()
        {
                Reporter.log("open the logs ",true);
        }

        @AfterClass
        publicvoid closeApp()
        {
                Reporter.log("close the app",true);
        }
        @BeforeMethod
        publicvoid login()
        {
                Reporter.log("Login ",true);
        }

        @AfterMethod
        publicvoid logout()
        {
                Reporter.log("logout",true);
        }

        @Test
        publicvoid createcustomer()
        {
                Reporter.log("Registercusotmer",true);
        }

        @Test
        publicvoid Deletecustomer()
        {
                Reporter.log("Delete coustomer",true);

        }
}
```

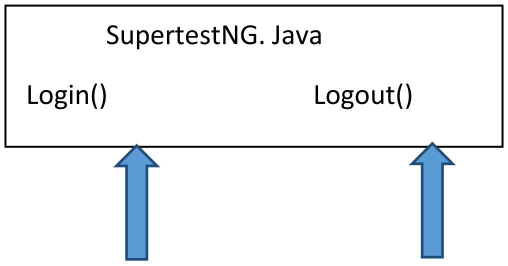
Note: Other Annotation of TestNG (http://testng.org/doc/documentation-main.html)

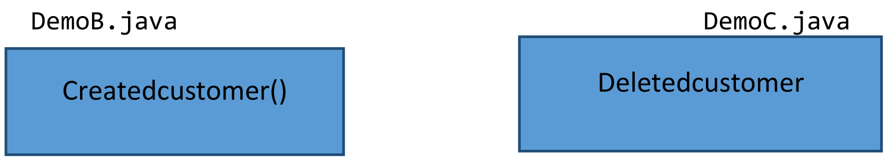
1) @BeforeSuite
2) @AfterSuite
3) @BeforeTest
4) @AfterTest
5) @BeforeGroups
6) @AfterGroups
7) @Factory
8) @Listener
9) @parameters
10)@Dataprovider

Inheritance in TestNG

While developing Automation scriot for every manual testcase we write TestNG class that is if we are autmomating 100 manual testcase we create 100 testNG class.

In every TestNG class ther may be common steps, such as Login and logout .instead of writing those methods again and again we can use inheritance concept. That is we develeop all the common methods in the parent class then inherted. In all the TestNG class as show in the below example.





(DemoB and DemoC inherited the supertestNG class)

```
---------------------------------------SupertestNG.java------------------------------
package com.qspider;

import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;

publicclass SupertestNG
{
	@BeforeMethod
	publicvoid login()
	{
		Reporter.log("Login ", true) ;

	}

	@AfterMethod
	publicvoid logout()
	{
		Reporter.log("Logout", true) ;
	}
}
  DemoB.java

package com.qspider;

import org.testng.Reporter;
import org.testng.annotations.Test;

publicclass DemoB extends SupertestNG
{
	@Test
	publicvoid createcustomer()
	{
		Reporter.log("Create customer", true) ;
	}



}

------------------------DemoC.java-----------------------------
package com.qspider;
```

```
import org.testng.Reporter;
import org.testng.annotations.Test;

publicclass demoC extends SupertestNG
{
	@Test
	publicvoid deletecustormer()
	{
		Reporter.log("Delete customer", true) ;

	}
}
```

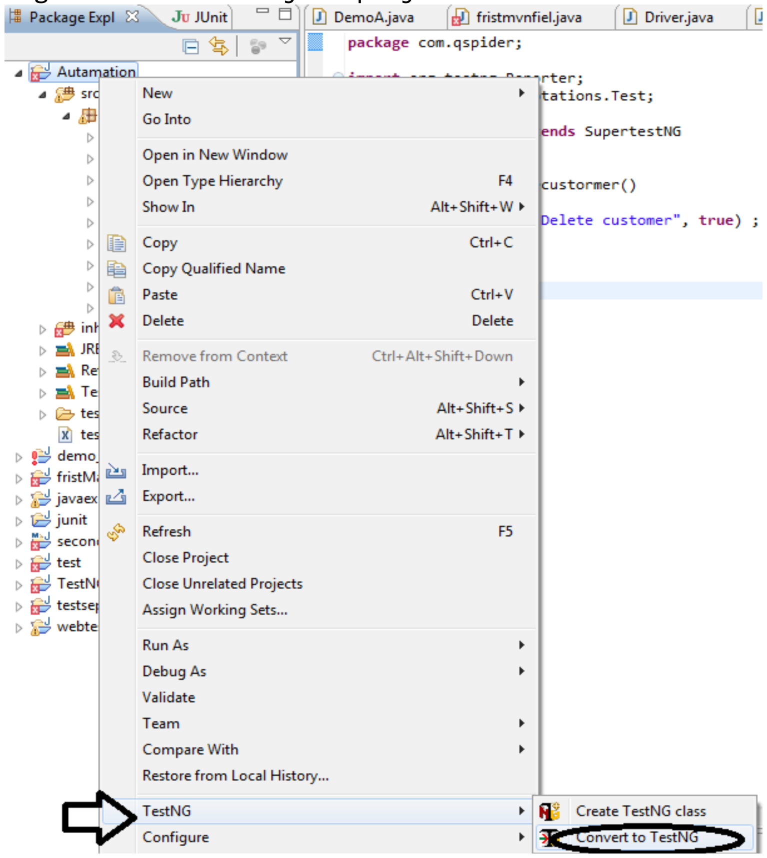
```
Output:
Login
Create customer
Logout
Login
Delete customer
Logout

===============================================
Suite
Total tests run: 2, Failures: 0, Skips: 0
===============================================
```

Test suite:
It is an xml file which contains list of all the TestNG class which are to be executed .

Creating the suite:

Vinay2ml@gmail.com

1) Right click on the java project ->TestNG→Convert to TestNG



```
package com.qspider;
rter;
tations.Test;
ends SupertestNG
custormer()
Delete customer", true) ;
```

2) It create TestNG.xml file inside the java project .Xml contains :

```
<suitename="Suite"parallel="none">
    <testname="Test">
    <classes>
            <classname="com.qspider.DemoB"/>
<classname="com.qspider.democ"/>
</classes>
</test>
</suite>
```

In order to execute the TestNG suite Right click on the TestNG.xml go to Run as Select TestNG suite.

1) Script execution is successful But still we want to fail the script explicitly ?
   Assert.fail()

Ex: **package** com.qspider;

```
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.Test;

publicclass democ extends SupertestNG
{
        @Test
```

```
	publicvoid deletecustormer()
	{
		Reporter.log("Delete customer", true) ;
		Assert.fail() ;
	}
}
```

2) How do you re-execute only the failed script, when ever ther is failure in the execution TestNG automatically creates Testng-failed.xml file inside the test-out floder which contains list of all the failed testNG class or method. So in order to execute only the failed script Right click on the xml file go to run as select TestNG suite.

3) How do you execute a method multiple times ?
   @Test(invocationCount =5)

   Ex:

```
package com.qspider;

import org.testng.Reporter;
import org.testng.annotations.Test;

publicclass DemoB extends SupertestNG
{

	@Test(invocationCount =5)
	publicvoid createcustomer()
	{
		Reporter.log("Create customer", true) ;
	}



	}
```

   Output:

```
Login
Create customer
Logout
Login
Create customer
Logout
Login
Create customer
Logout
Login
Create customer
Logout
```

```
Login
Create customer
Logout
PASSED: createcustomer
PASSED: createcustomer
PASSED: createcustomer
PASSED: createcustomer
PASSED: createcustomer

===============================================
    Default test
    Tests run: 5, Failures: 0, Skips: 0
```

4) How do you execute a method multiple times with different input?
   dataprovider=”logindata”

   ex:

```
package com.qspider;

import org.testng.Reporter;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;

publicclass dataproviderex {

	@DataProvider(name ="logindata")
	public Object[][] getData()
	{
		Object[][] data = new Object[2][2] ;
		data[0][0] ="admin" ;
		data[0][1] = 1234 ;
		data[1][0] ="userA" ;
		data[1][1] =3455;
		return data ;

	}

	@Test(dataProvider="logindata")
	publicvoid loginlogout(String un, int pin)
	{
		Reporter.log("enter un "+un, true) ;
		Reporter.log("Enter pin" +pin, true) ;
		Reporter.log("Click login ",true) ;
		Reporter.log("click logout", true) ;
	}
}
```

   Output:

```
enter un admin
Enter pin1234
Click login
click logout
```

```
enter un userA
Enter pin3455
Click login
click logout
PASSED: loginlogout("admin", 1234)
PASSED: loginlogout("userA", 3455)

===============================================
    Default test
    Tests run: 2, Failures: 0, Skips: 0
      ===============================================
```

Note: in automation testing we use XL file as a source to test the application with multiple inputs (data driven testing) because dataprovider as following issues :

1) Size of the data is fixed
2) Maintaining large test data is very difficult ex: 100 users
3) We can’t reuse the data provider, if it present in the different class

Parallel execution using TestNG:

While performing browser compatibility testing execute the same script on multiple browser, but of execute in sequential order , it will time consuming , instead of this we can use TestNG to execute the script in parallel.

Content of the TestNG class:

```
package com.qspider;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;

publicclass DemoD {

	@Parameters({"browser"})
	@Test
	publicvoid testD(String browser)
	{
		WebDriver driver ;
		if(browser.equals("GC"))
		{
			System.setProperty("webdriver.chrome.driver", 
"C:/selenium/Selenium-Dump/chromedriver.exe") ;
```

```
                    driver = new ChromeDriver() ;

             }
             else
             {
                    driver = new FirefoxDriver() ;

             }

      driver.manage().timeouts().implicitlyWait(20,TimeUnit.SECONDS) ;
      driver.get("http://demo.actitime.com/") ;
      for(int i =1 ;i <=100 ; i++)
      {
             driver.findElement(By.id("username")).sendKeys("vinayjugga") ;
             driver.findElement(By.id("username")).clear();
      }
 driver.close() ;
      }

}
```

Note: we can't execute above class directly because testMethod is taking the parameter we should always run this from the TestNG suite.

Testng.xml

```
<suitename="Suite"parallel="tests">
<testname="GCTest">
<parametername="browser"value="GC"/>
<classes>

<classname="com.qspider.DemoD"/>
</classes>
</test><!-- Test -->


<testname="FFTest">
<parametername="browser"value="FF"/>
<classes>

<classname="com.qspider.DemoD"/>
</classes>
</test><!-- Test -->
</suite><!-- Suite -->
```

Encapsulation: binding the data member and member function together in a secure way is called encapsulation.

Ex:

```
package com.qspider;
```

```
import java.sql.Driver;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;

publicclass encapsulationex {

       private WebElement un ;
       private WebDriver driver ;

       publicdemo()
        {
              un = driver.findElement(By.id("usename")) ;
        }

       publicvoid setter(String username)
        {
             un.sendKeys(username);
        }
          }
```

Automation frame work:

It is a set of Rules guidelines and best practice to automated application in automation frame work. There are 3 stages.

1) Framework Design
2) Framework Implementation
3) Framework Execution

The above stages depends on type of the framework which is used in the automation.

We can broadly categorized the framework types as specified below irrespective of the automation tools.

1) Linear automation Framework
2) Data driven automation Framework
3) Method driven automation Framework it is also called as Function drivern or action driven
4) Module driven Automation framework
5) Keyword driven Automation Framework
6) Hybrid Automation Framework

Note: with respect to selenium there are popular frameworks which are implemented using any one of the above types.

Ex: Page object module (POM)

Robot framework, cucumber framework.

Framework Design: Senior automation engineer or leads with their past experience will design the automation framework.

1) They list of out of all the software files which are required for the automation and then specify the folder structure and their purpose.
2) Required software:
   1) JDK
   2) Ecliepse +TestNG plugin
   3) Firefox +Firebug +Firepath plugin
   4) AutoIt
   5) Google chrome browser
   6) MS office

Note: ensure that all the mentioned software are installed and there are updated .

Required files:

1) Selenium server stand alone .jar file
2) Apache and POI jar files
3) Chromedriver.exe file
4) IDserver.exe file

Configuring the Framework:

1) Create floder c:\AWS
2) In Eclipse IDE go to file->Switch workspace and select the AWS folder
3) Create a new Java project –specify the name ex: Automation

In the automation framework we will be using the below files:

1) .jar
2) .exe
3) .xls
4) .xml
5) .html
6) .bat
7) .class
8) .java

Note: Except first 3 types of files all the other files will have predefined location such as :

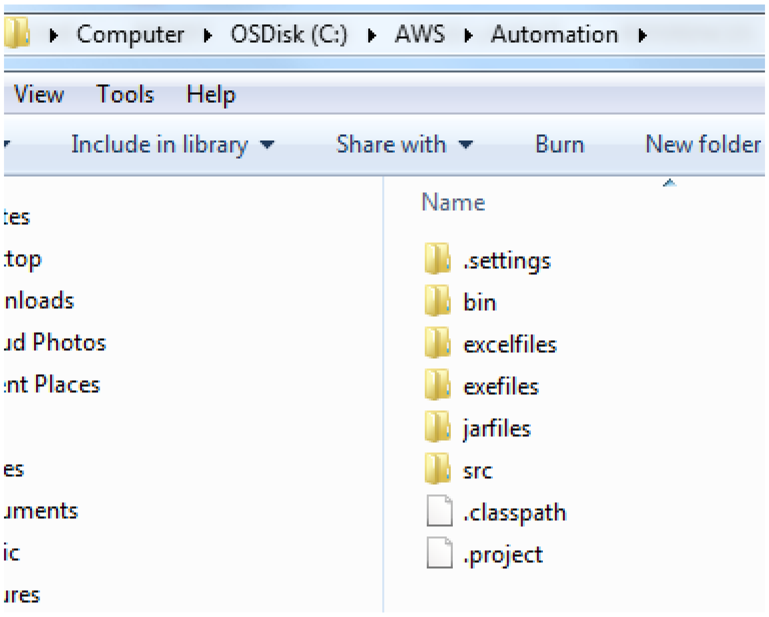
Src : .java files

Bin: .class files etc..

In the java project folder we should create 3 floders to store the first 3 types of files.

C:\AWS\Automation

Create the folders : ex: Exefiles, Excelfiles, Jarfiles.

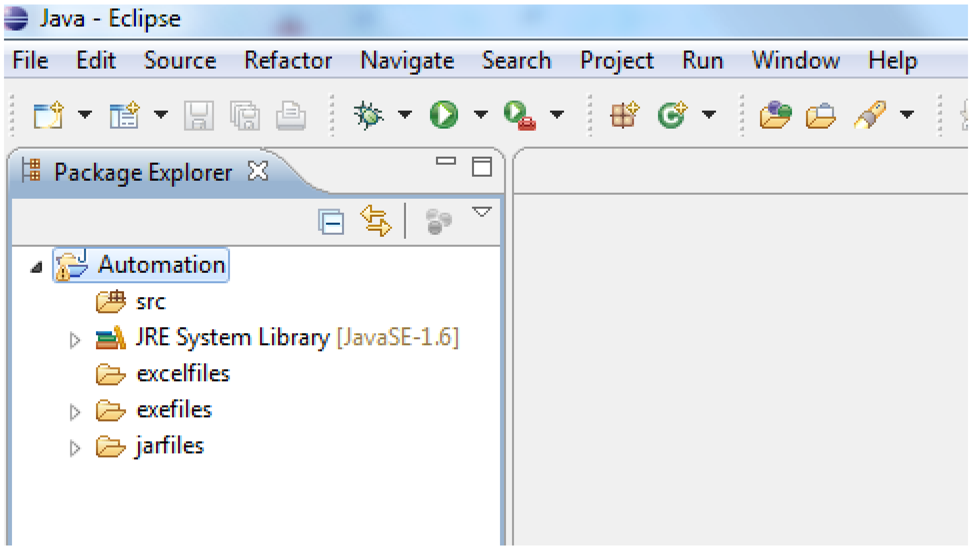


1) Copy the selenium jar files into jarfile floder and chromedriver.exe and IEdriver.exe copy paste into the exefiles.

Note: if we write any AuotIT script the framework implementation then autoIt script also stored into the exefiles.

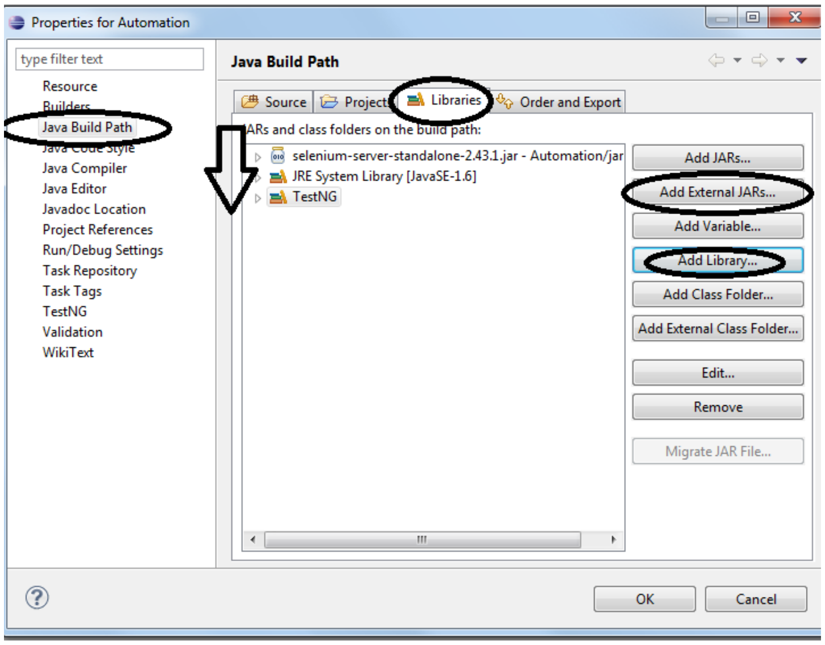
Simllarly if we any xlfiles for the test data it will be copy paste into the excelfiles folder.

In eclipse IDE right click on the java project and Refresh which will display newly created floder.


Java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Automation
src
JRE System Library [JavaSE-1.6]
excelfiles
exefiles
jarfiles

Select all the jar files in the Java build path select library stack and click Add external Jar files

Click on Add library select TestNg click finish



Implementation of the framework:

This is the actual stage where we develop the required Methods and convert the Manual test case's into automation Script.

Java Design pattern:

Before writing any code developer will consult architecture for the design of the application in java 250 + Design solution are ready available which are called as java design pattern .

Some of the famous and frequent used java design pattern are

1) Signleton
2) Multiton
3) Factory
4) Page object Module

For developing the web application page object model is best suitable.

Page object Module:

This is the one of the java design pattern where the below rules to develop the webpage.

1) For every webpage present in the application developer develops 1 Java class.
2) Every java class should contain data members and constructor and member function
3) Data member represent the elements present on the webpage and all the data member should be private.
4) Constructor is used to initialize the element present on the webpage this process is called as page decoration. In page object module we use parameter side constructor.
5) All the method should be public and they are used to perform action on the element in automation also we use page object module concept to test the webpage.

Ex:

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;


publicclass loginpage
{
	private WebElement untexbox ;

	public loginpage(WebDriver driver)
	{
		untexbox = driver.findElement(By.id("username")) ;
	}

	publicvoid setUserName(String un)
	{
		untexbox.sendKeys(un) ;
	}

	public String getUserName()
	{
		returnuntexbox.getAttribute("value");
	}
}
```

Abstraction: hiding the implementation from the end user generally called as abstraction .

We use this concept to reduce the burden on the end user and also reduce the maintaince.

Ex: Instead of developing setter and getter method for every field we develop the method based on the functionality

Ex: instead of developing 3 method to handle username and password and login button sparetly , we develop method as shown below .

```
publicclass loginpage
{
	private WebElement untexbox ;
	private WebElement loginbutton;

	publicvoid login(String un, String pw)
	{
		untexbox.sendKeys(un) ;
		untexbox.sendKeys(pw) ;
		loginbutton.click() ;
	}

}
```

The end user just call a method to perform the login operation on the application without knowing the actual implementation of the code.

Ex:

```
import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;




publicclass loginusage {

	publicstaticvoid main(String[] args) {
		WebDriver driver  = new FirefoxDriver() ;
		driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS) ;
		driver.get("http://demo.actitime.com/") ;
		LoginPage loginpage = new LoginPage(driver) ;
		loginpage.login("admin", "manager");

	}

}
```

Page Factory:

It is a selenium class which implements page object module concept in selenium.

Generally when we try to initialize the element in the constructor at the beginning only so we may get exception, because the element may not be present at that time.

Ex:

Error msg is displayed only entering after the criditianls popup is displayed only click in on after click login button get exception.

While handling the very large webpage where it has too many elements will drastically increase total length of the constructor.

Loading all the elements into the memory at a time will affect performance of the script also to overcome the above issue we use Page factory.

In page factory we should declare the element using @FindBy Annoation and we use initElements() methods to initialize all these elements during runtime instead of the initial time this process is called as late binding or lassie initialization

Ex:

```
importorg.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;


      publicclass LoginPage
      {
            @FindBy(id ="username")
            private WebElement untexbox ;

            @FindBy(name ="pwd")
            private WebElement passwd;

            @FindBy(id ="loginButton")
            private WebElement loginbutton ;

            @FindBy(className ="errormsg")
            private WebElement emsg;


            public LoginPage(WebDriver driver)
            {
                  PageFactory.initElements(driver, this);
            }

            /*public LoginPage(WebDriver driver)
            {
                  untexbox = driver.findElement(By.id("username")) ;
                  untexbox = driver.findElement(By.name("pwd"));
                  loginbutton =driver.findElement(By.id("loginButton"));
            }
 */
            /*public void setUserName(String un)
            {
                  untexbox.sendKeys(un) ;
            }

            public String getUserName()
            {
                  return untexbox.getAttribute("value");
            }
*/

            publicvoid login(String un, String pw)
            {
                  untexbox.sendKeys(un) ;
                  passwd.sendKeys(pw) ;
                  loginbutton.click() ;
            }

            public String printErrorMsg()
            {

                  String emsg1 = emsg.getText() ;
                  System.out.println(emsg1);
                  return emsg1;
            }
      }

mport java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;



publicclass loginusage {

      publicstaticvoid main(String[] args) {
            WebDriver driver  = new FirefoxDriver() ;
            driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS) ;
            driver.get("http://demo.actitime.com/") ;
            LoginPage loginpage = new LoginPage(driver) ;
            loginpage.login("admin", "manager1");
            loginpage.printErrorMsg();


      }

}
```

How do you handle multiple elements using FindBy Annoation

@FindBy(xpath =”//a”)

Private List<WebElement> alllinks;

Note: The above statement equivalent to findElements() methods..

What is the limitation of findBy Annotation:

@FindBy we can’t use variable in FindBy Annotation. While specifying the value of the locator .

```
Ex: String xp =”//a”
@FindBy(xpath =xp)   //Error
Private webElement link:
```

Developing Page object Module:

Developing Sample test case :http://demo.actitime.com/

1) Testcase_ID :ActiveTime_LoginLogout()
   Steps:
   - ➔ Enter valid username
   - ➔ Enter valid password
   - ➔ Enter on login button
   - ➔ Click on logout link
2) TestCase_ID: ActiveTime_CreateBilling()
   Steps:
   - →Enter valid user name and password
   - →Click on login button
   - →Click on setting and then click on Billing types
   - →click on Create Billing Types
   - →Enter the name ex: Automation and click on create Billing Type
   - →Verify that following success Message displayed
     Msg =”Billing type has been successfully created”
   - →Click on logout
3) TestCase ID: ActiveTime_DeleteBilling()
   Steps:
   - ➔ Login to activetime with valid username and password
   - ➔ Click on settings and click on billing types
   - ➔ Click on delete link on Autoatmion click ok on confirmation pop-up
   - ➔ Verify that following msg is displayed
   - ➔ Msg=”Billing type has been successfully deleted.
   - ➔ Click on logout link

Note:

1) Execute the testcase manually at least once , its given more clarity on the testcase should be automated
2) For the given testcase note down the elements and action perform on those Elements based on the webpages
3) Some of the elements will be present in multiple webpages such as headers and footers.

   Ex: Time track menu , Settings, helps, logout etc.

Instead of storing and writing method in every class we use inheritance concept. That is we created superclass, which is called as Basepage and then we make all other class in the page object module to extends Base page class, except login page.

1) Pages and Class:

   Page1: LoginPage
   Elements: usernametextbox, pwtextbox, loginbutton
   Methods:
    Method1-------login()
      → Enter un
        ➔ Enter pw
        ➔ Click login

```
importorg.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;


      publicclass LoginPage
      {
             @FindBy(id ="username")
             private WebElement untexbox ;

             @FindBy(name ="pwd")
             private WebElement passwd;

             @FindBy(id ="loginButton")
             private WebElement loginbutton ;

             @FindBy(className ="errormsg")
             private WebElement emsg;


             public LoginPage(WebDriver driver)
             {
                    PageFactory.initElements(driver, this);
             }


             publicvoid login(String un, String pw)
             {
                    untexbox.sendKeys(un) ;
                    passwd.sendKeys(pw) ;
                    loginbutton.click() ;
             }
```

```
        }

           Page2: BasePage

          Elements: Settings,billingTypes, Logoutlink

         Methods:

                Method1:----gotobillingpage()

                  ➔ Click on settings
                  ➔ Click on billing types

                Method2: Logout()

                      →Click on logout link
```

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;


publicclass BasePage {

	@FindBy(xpath ="(//div[@class ='popup_menu_arrow'])[1]")
	private WebElement settings ;

	@FindBy(linkText ="Billing Types")
	private WebElement billingTypes;

	@FindBy(id ="logoutLink")
	private WebElement logoutLink;

	public BasePage(WebDriver driver) {
		PageFactory.initElements(driver,this);
	}


	publicvoid gotoBillingPage()
	{
		settings.click();
		billingTypes.click();
	}


	publicvoid logOut()
	{
		logoutLink.click();
	}


}
```

Page3 : createnewBillingpage

Elements: name, createBillingTypebutton

Method1: CreateBilling()

→Enter name

→Click createBillingType button

```java
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;


publicclass CreateNewBillingPage extends BasePage
{
	public CreateNewBillingPage(WebDriver driver) {
		super(driver);
		PageFactory.initElements(driver, this);
	}

	@FindBy(id="name")
	private WebElement name;

	@FindBy(xpath="//input[contains(@value,'Create Billing Type')]")
	private WebElement createBillingTypeButton ;

	publicvoid createBilling(String billname)
	{
		name.sendKeys(billname);
		createBillingTypeButton.click();
	}



}
```

Page4: BillingTypepages

Elements: CreateBillingTypeButton, msg1, msg2, deletelink, ok button(pop-up)

Methods:

Methods1: clickcreateBillingTypes

→Click on createBilling Types button

Method2: verify msg1

Method2: verify msg2

Method4: Delete Billing

1) Click on delete link
2) Click ok on the pop-up

```java
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.testng.Assert;


publicclass BillingTypesPage extends BasePage {

	@FindBy(xpath ="//span[text()='Create Billing Type']")
	private WebElement createBillingtypesbutton ;

	@FindBy(xpath ="//span[@class='successmsg']")
	private WebElement amsg ;

	@FindBy(xpath ="//a[text()='automation']/../../td[6]/a")
	private WebElement deleteLink ;


	public BillingTypesPage(WebDriver driver)
	{
		super(driver);
		PageFactory.initElements(driver, this);
	}


	private WebElement okbutton;

	publicvoid clickCreateBillingTypes()
	{
		createBillingtypesbutton.click();
	}

	publicvoid verifymsg(String emsg)
	{
	  String actualmsg =amsg.getText();
	  Assert.assertEquals(actualmsg, emsg);

	}

	publicvoid deleteBilling(WebDriver driver)
	{
		deleteLink.click();
		driver.switchTo().alert().accept() ;
	}



}
```

Testcase_ID :ActiveTime_LoginLogout()

Steps:

- ➔ Enter valid username
- ➔ Enter valid password
- ➔ Enter on login button
- ➔ Click on logout link

```
import org.testng.annotations.Test;


publicclass testActiveTime_LoginLogout extends SuperTestNG {

	@Test
	publicvoidtestActiveTime_LoginLogout()
	{
	LoginPage loginpage = new LoginPage(driver) ;
	loginpage.login("admin", "manager");

	BasePage basepage =new BasePage(driver);
	basepage.logOut() ;

	}

}
```

TestCase_ID: ActiveTime_CreateBilling()

Steps:

- →Enter valid user name and password
- →Click on login button
- →Click on setting and then click on Billing types
- →click on Create Billing Types
- →Enter the name ex: Automation and click on create Billing Type
- →Verify that following success Message displayed
  Msg =”Billing type has been successfully created”
- →Click on logout

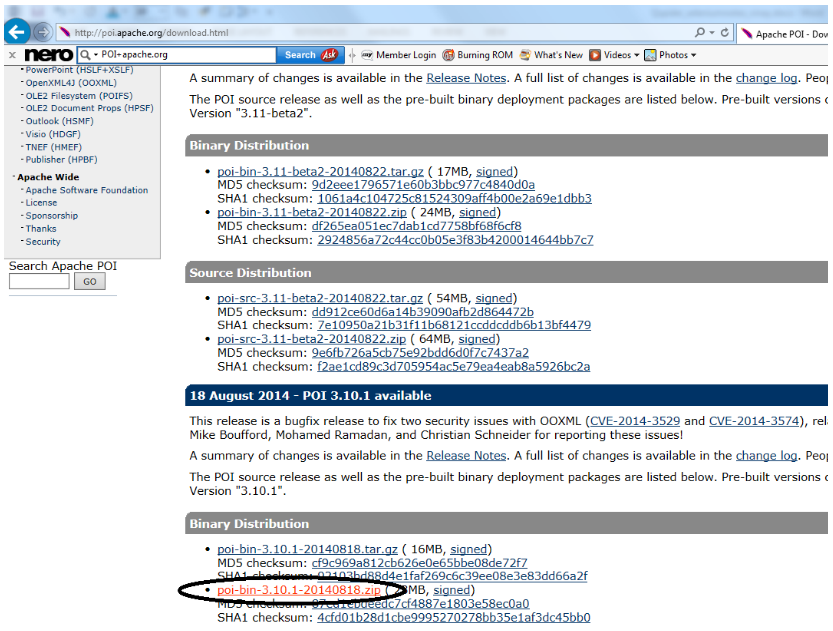
```
import org.testng.annotations.Test;


publicclass testActiveTime_createBilling extends SuperTestNG {

	@Test
	publicvoidtestActiveTime_createBilling(){
        LoginPage loginpage = new LoginPage(driver) ;
        loginpage.login("admin", "manager");

        BillingTypesPage billpage = new BillingTypesPage(driver);

        billpage.gotoBillingPage() ;
        billpage.clickCreateBillingTypes();

        CreateNewBillingPage  createbill  = new CreateNewBillingPage(driver);
        createbill.createBilling("Automation");

        billpage.verifymsg("Billing type has been successfully created.");

        billpage.logOut() ;

        }
}
```

Data Driver Test:

Testing the application with multiple inputs is called as Data driver testing in order to do this we take the test data from the Excel sheet.

In order to handle the excel sheet we use POI (Poor obfuscation implementation) jar file provided by Apache which can be downloaded from the following website.

http://poi.apache.org/download.html



A summary of changes is available in the Release Notes. A full list of changes is available in the change log. Peo

The POI source release as well as the pre-built binary deployment packages are listed below. Pre-built versions
Version "3.11-beta2".

Binary Distribution

- poi-bin-3.11-beta2-20140822.tar.gz (17MB, signed)
  MD5 checksum: 9d2eee1796571e60b3bbc977c4840d0a
  SHA1 checksum: 1061a4c104725c81524309aff4b00e2a69e1dbb3
- poi-bin-3.11-beta2-20140822.zip (24MB, signed)
  MD5 checksum: df265ea051ec7dab1cd7758bf68f6cf8
  SHA1 checksum: 2924856a72c44cc0b05e3f83b4200014644bb7c7

Source Distribution

- poi-src-3.11-beta2-20140822.tar.gz (54MB, signed)
  MD5 checksum: dd912ce60d6a14b39090afb2d864472b
  SHA1 checksum: 7e10950a21b31f11b68121ccddcddb6b13bf4479
- poi-src-3.11-beta2-20140822.zip (64MB, signed)
  MD5 checksum: 9e6fb726a5cb75e92bdd6d0f7c7437a2
  SHA1 checksum: f2ae1cd89c3d705954ac5e79ea4eab8a5926bc2a

18 August 2014 - POI 3.10.1 available

This release is a bugfix release to fix two security issues with OOXML (CVE-2014-3529 and CVE-2014-3574), rel
Mike Boufford, Mohamed Ramadan, and Christian Schneider for reporting these issues!

A summary of changes is available in the Release Notes. A full list of changes is available in the change log. Peo

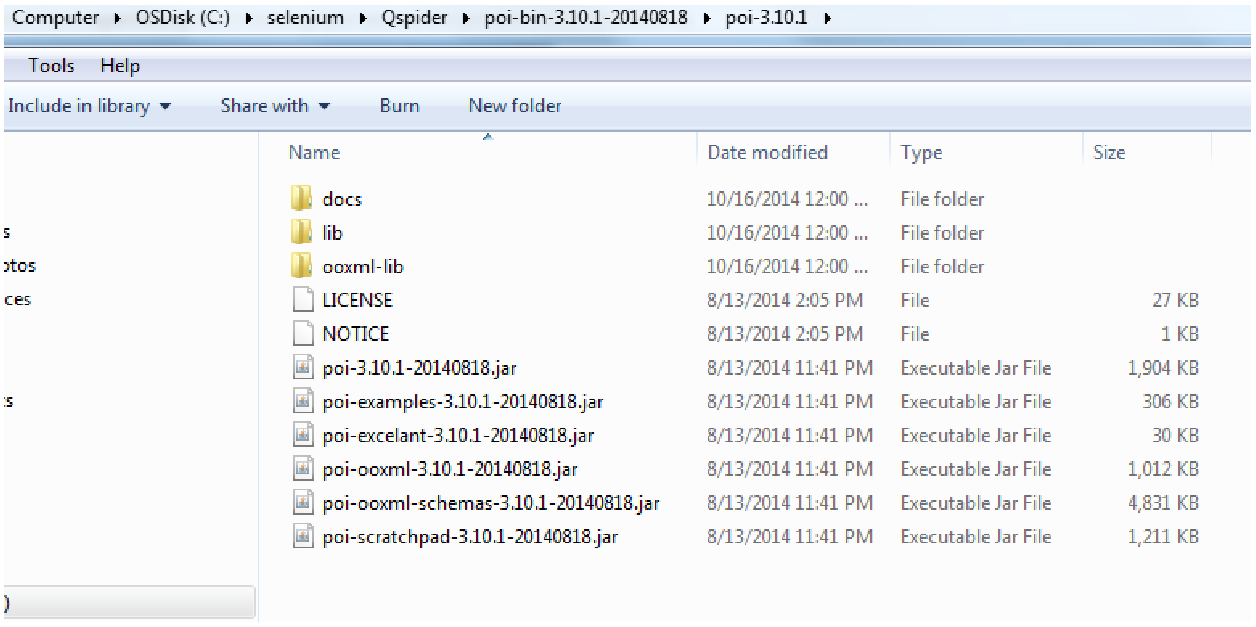
The POI source release as well as the pre-built binary deployment packages are listed below. Pre-built versions
Version "3.10.1".

Binary Distribution

- poi-bin-3.10.1-20140818.tar.gz (16MB, signed)
  MD5 checksum: cf9c969a812cb626e0e65bbe08de72f7
  SHA1 checksum: 02103bd88d4e1faf269c6c39ee08e3e83dd66a2f
- poi-bin-3.10.1-20140818.zip ([illegible]MB, signed)
  MD5 checksum: [illegible]edc7cf4887e1803e58ec0a0
  SHA1 checksum: 4cfd01b28d1cbe9995270278bb35e1af3dc45bb0

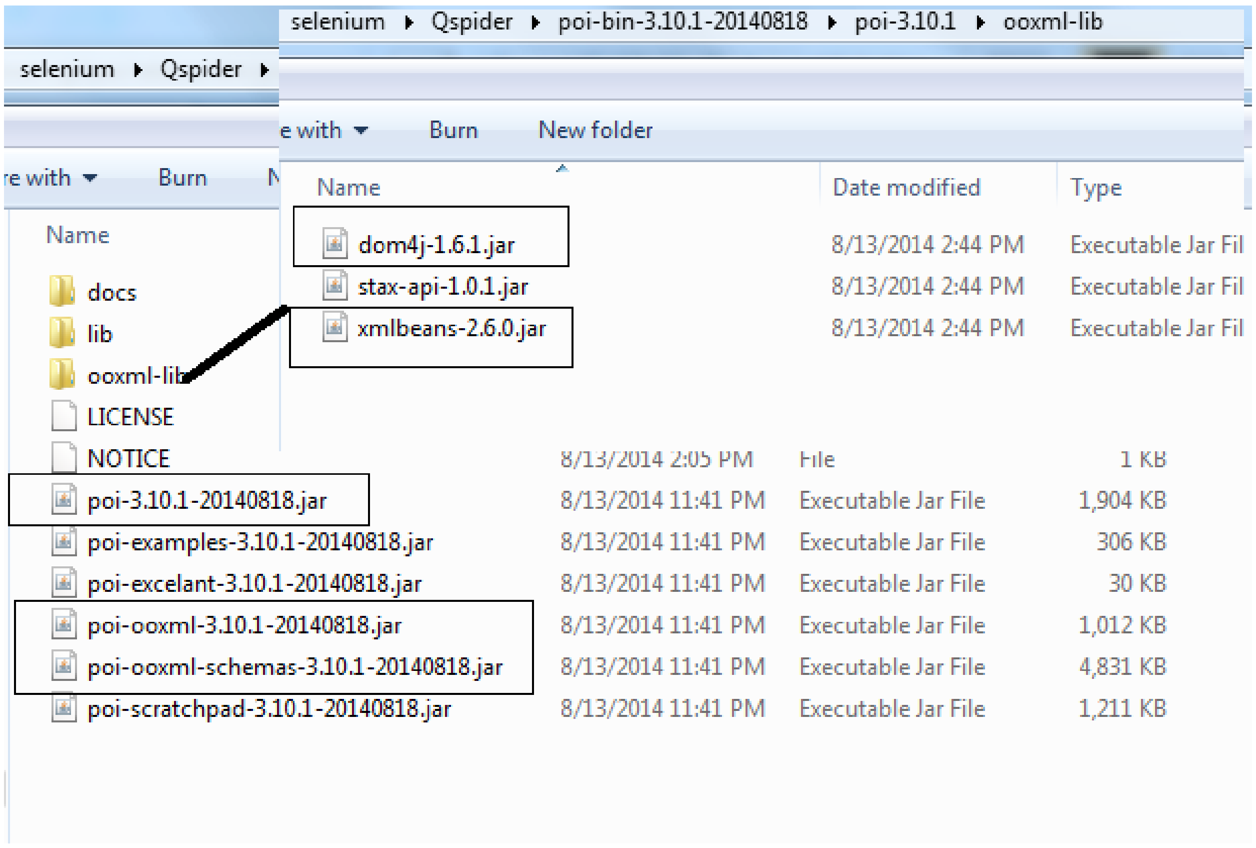
Download the zip file and copy paste the downloaded file into required location and unzip.

It will create the folder with the following:


Computer ▸ OSDisk (C:) ▸ selenium ▸ Qspider ▸ poi-bin-3.10.1-20140818 ▸ poi-3.10.1 ▸
Tools
Help
Include in library
Share with
Burn
New folder

Name	Date modified	Type	Size
docs	10/16/2014 12:00 ...	File folder	
lib	10/16/2014 12:00 ...	File folder	
ooxml-lib	10/16/2014 12:00 ...	File folder	
LICENSE	8/13/2014 2:05 PM	File	27 KB
NOTICE	8/13/2014 2:05 PM	File	1 KB
poi-3.10.1-20140818.jar	8/13/2014 11:41 PM	Executable Jar File	1,904 KB
poi-examples-3.10.1-20140818.jar	8/13/2014 11:41 PM	Executable Jar File	306 KB
poi-excelant-3.10.1-20140818.jar	8/13/2014 11:41 PM	Executable Jar File	30 KB
poi-ooxml-3.10.1-20140818.jar	8/13/2014 11:41 PM	Executable Jar File	1,012 KB
poi-ooxml-schemas-3.10.1-20140818.jar	8/13/2014 11:41 PM	Executable Jar File	4,831 KB
poi-scratchpad-3.10.1-20140818.jar	8/13/2014 11:41 PM	Executable Jar File	1,211 KB

Copy only below mentioned 5 jar file into the Jar file folder into the automation framework.

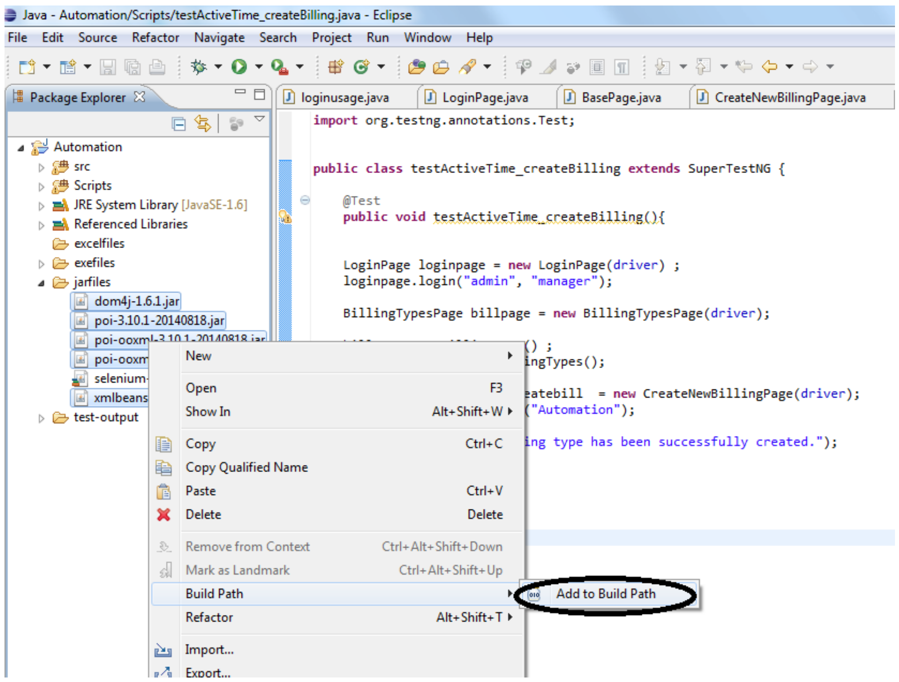

selenium ▸ Qspider ▸ poi-bin-3.10.1-20140818 ▸ poi-3.10.1 ▸ ooxml-lib
selenium ▸ Qspider ▸
e with
Burn
New folder

Name	Date modified	Type
dom4j-1.6.1.jar	8/13/2014 2:44 PM	Executable Jar Fil
stax-api-1.0.1.jar	8/13/2014 2:44 PM	Executable Jar Fil
xmlbeans-2.6.0.jar	8/13/2014 2:44 PM	Executable Jar Fil

Name	Date modified	Type	Size
docs			
lib			
ooxml-lib			
LICENSE			
NOTICE	8/13/2014 2:05 PM	File	1 KB
poi-3.10.1-20140818.jar	8/13/2014 11:41 PM	Executable Jar File	1,904 KB
poi-examples-3.10.1-20140818.jar	8/13/2014 11:41 PM	Executable Jar File	306 KB
poi-excelant-3.10.1-20140818.jar	8/13/2014 11:41 PM	Executable Jar File	30 KB
poi-ooxml-3.10.1-20140818.jar	8/13/2014 11:41 PM	Executable Jar File	1,012 KB
poi-ooxml-schemas-3.10.1-20140818.jar	8/13/2014 11:41 PM	Executable Jar File	4,831 KB
poi-scratchpad-3.10.1-20140818.jar	8/13/2014 11:41 PM	Executable Jar File	1,211 KB

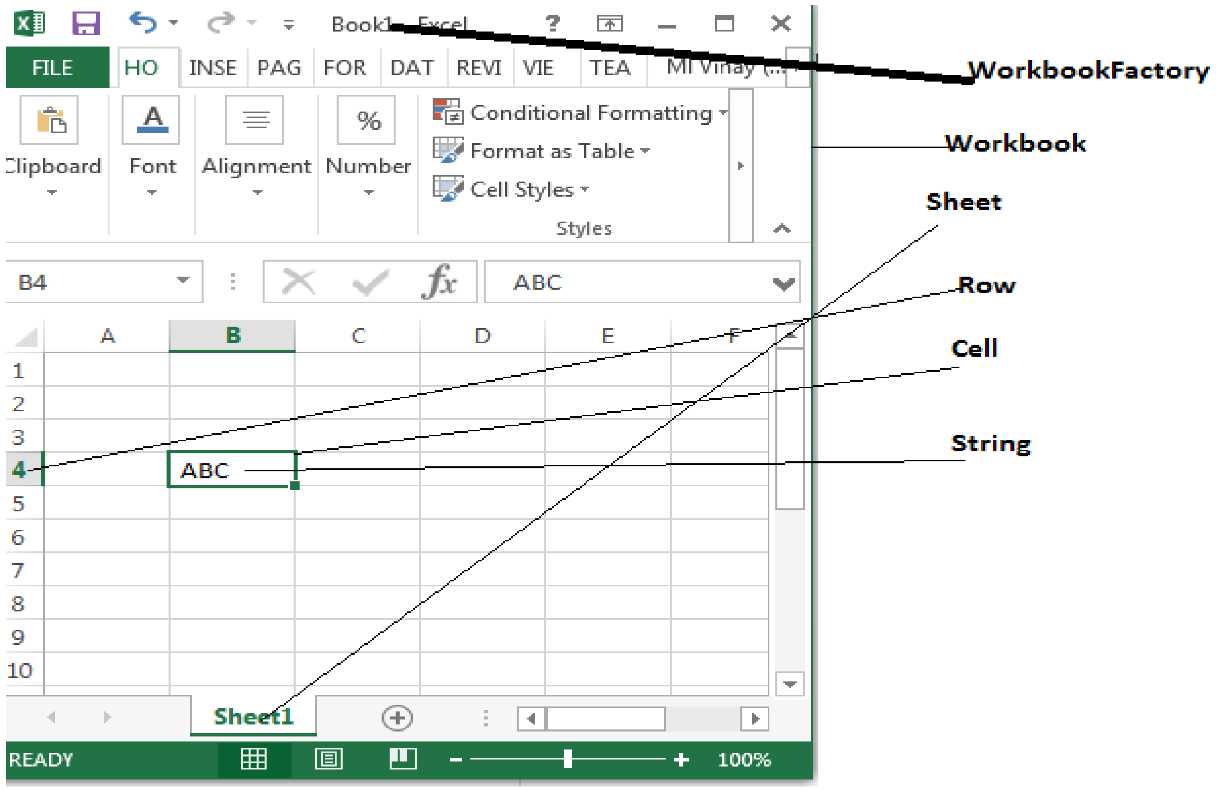
Vinay2ml@gmail.com

Associate all the above selected jar files into the java project.



```
import org.testng.annotations.Test;

public class testActiveTime_createBilling extends SuperTestNG {

    @Test
    public void testActiveTime_createBilling(){

    LoginPage loginpage = new LoginPage(driver) ;
    loginpage.login("admin", "manager");

    BillingTypesPage billpage = new BillingTypesPage(driver);
                                                    () ;
                                                    ingTypes();

                                                    eatebill  = new CreateNewBillingPage(driver);
                                                    ("Automation");

                                                    ing type has been successfully created.");
```

Basic architecture of Excel with respect to Apache POI files



WorkbookFactory -----------Read ---→FileinputStream

```
          Workbook  -----------------write--→ outputstream

          In Excel Row No and Cellno start with Zero(0).
          Ex: Reading data present in the first cell of the Xl sheet1 0 and cell 0.

          Ex:
```

```
import java.io.FileInputStream;
importjava.io.FileNotFoundException;
import java.io.IOException;

import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;



publicclass ExcelDemo {

	publicstaticvoid main(String[] args) throws IOException, 
InvalidFormatException
	{
		String xlpath ="C:/AWS/Automation/excelfiles/login.xlsx" ;
		FileInputStream filein = new FileInputStream(xlpath);
		Workbook wb = WorkbookFactory.create(filein);
		Sheet s1 = wb.getSheet("sheet1");
		Row r = s1.getRow(0) ;
		Cell c =r.getCell(0);
		String v = c.getStringCellValue();
		System.out.println(v);

	}

}
```

	Optimized code:

```
import java.io.FileInputStream;
importjava.io.FileNotFoundException;
import java.io.IOException;

import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;


publicclass ExcelDemo1 {

	publicstaticvoid main(String[] args) throws IOException, 
InvalidFormatException {
```

```
            //.(dot) represent the JAVA Path ex: C:/AWS/Automation/
            String xlpath = "./excelfiles/login.xlsx" ;
            FileInputStream filein =new FileInputStream(xlpath) ;
            Workbook wb =WorkbookFactory.create(filein);
            Sheet s1 = wb.getSheet("sheet1") ;
            String v = s1.getRow(0).getCell(0).getStringCellValue();
            System.out.println(v);

      }

}
```

Note:

1) Instead of the specify complete path of the XL file absolute path we can specify Relative path using which represents the complete path of current JAVA project.
2) FileInputStream will throw IOException (Checked exception)
   Ex:
   ```
   Exception in thread "main" java.io.FileNotFoundException:
   .\excelfiles\login1.xlsx (The system cannot find the file specified)
   ```

3) Create Factory method will throw InvalidFormatException it also(checked Exception)
   ```
         Ex: String xlpath = "./excelfiles/login.docx" ;
   Exception in thread "main" java.lang.IllegalArgumentException: Your
   InputStream was neither an OLE2 stream, nor an OOXML stream
   ```

4) If sheet name or roll number or cell number is invalid will get NuLLpointerException.(unchecked Exception)

   ```
   Ex:String v = s1.getRow(100).getCell(0).getStringCellValue();
   Exception in thread "main" java.lang.NullPointerException
   ```

1) Write a code to count the number of Rows present in the Xl sheet.

```
import java.io.FileInputStream;
importjava.io.FileNotFoundException;
import java.io.IOException;

import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;


publicclass RowCount {


      publicstaticvoid main(String[] args) throws IOException,
InvalidFormatException
      {
```

```
		String xlpath ="C:/AWS/Automation/excelfiles/login.xlsx" ;
		FileInputStream filein = new FileInputStream(xlpath);
		Workbook wb =WorkbookFactory.create(filein);
		Sheet s1 = wb.getSheet("sheet1") ;
		System.out.println(s1.getLastRowNum());
		}

}
```

Note: getLastRowNum() return index of the last Row that is if the 10 rows are present , it will return 9.

2) Write a script to count the number of cells present in the first Row.

```
import java.io.FileInputStream;
importjava.io.FileNotFoundException;
import java.io.IOException;

import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;


publicclass CountCellfristRow {

	publicstaticvoid main(String[] args) throws IOException, 
InvalidFormatException {
		String xlpath = "./excelfiles/login.xlsx" ;
		FileInputStream  filein =new FileInputStream(xlpath);
		Workbook wb =WorkbookFactory.create(filein);
		Sheet s1 =wb.getSheet("sheet1");
		System.out.println(s1.getRow(2).getLastCellNum());



	}

}
```

Note: getLastCellNum() return the number of cell present in the specified Row and not the Index.

3) Write a code to print number of cells present in the each row of the excel sheet.

```
import java.io.FileInputStream;
importjava.io.FileNotFoundException;
import java.io.IOException;

import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;


publicclass CountEachRowCell {

	publicstaticvoid main(String[] args) throws IOException,
InvalidFormatException {
		String xlpath = "./excelfiles/login.xlsx" ;
		FileInputStream filein =new FileInputStream(xlpath);
		Workbook wb =WorkbookFactory.create(filein);
		Sheet s = wb.getSheet("sheet1");
		int countrow = s.getLastRowNum();
		for(int i =0 ; i<=countrow; i++)
		{
			System.out.println(s.getRow(i).getLastCellNum()) ;
		}


	}

}
```

4) Write the code to print the content of the xl Sheet.

```
import java.io.FileInputStream;
importjava.io.FileNotFoundException;
import java.io.IOException;

import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;


publicclass PrintContentXL {

	publicstaticvoid main(String[] args) throws IOException,
InvalidFormatException {
		String xlpath = "./excelfiles/login.xlsx" ;
		FileInputStream filein = new FileInputStream(xlpath);
		Workbook wb =WorkbookFactory.create(filein);
		Sheet s1 =wb.getSheet("sheet1");
		int rowcount = s1.getLastRowNum();
		for(int i =0; i<rowcount; i++)
		{
			Row r = s1.getRow(i) ;
			int countcell =r.getLastCellNum() ;
			for(int j=0; j <countcell ;j++)
			{
				System.out.print(r.getCell(j).getStringCellValue()+ " ") ;
```

```
                    }
                    System.out.println(" ");
            }




        }

}
```

Note: if any row and cell is blank in between then we get NullpointerException , we should handle it using try catch block.

5) Write data back to Excel Sheet .

   In order to store the value in the xl sheet we use SetCellValue() to Save the XL file use write() method of workbook, which takes FileoutputStream as argument as show in below

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;


publicclass WritetoExcel {

        publicstaticvoid main(String[] args) throws IOException,
InvalidFormatException {
                String xlpath = "./excelfiles/login.xlsx" ;
                FileInputStream filein =new FileInputStream(xlpath);
                Workbook wb =WorkbookFactory.create(filein);
                Sheet s = wb.getSheet("sheet1");
                //Existing row and cell to write new value
                s.getRow(0).getCell(0).setCellValue("java");

                //Existing row and new cell value
                s.getRow(0).createCell(17).setCellValue("Vinay");

                s.createRow(10).createCell(0).setCellValue("Jugga");

                FileOutputStream fos = new FileOutputStream(xlpath);
                wb.write(fos);
        }

}
```

Generic Methods:

Any Method Which can be used in different projects or called as Generic methods. Such as handling XL files and handling Database etc.

Any Method which can be used only with in the current project or called as Project specific method.

Ex: All the methods develop under page object module, ex: createBilling, DeleteBilling etc.

Steps to Develop Generic Library:

1) Crate package with the name .com.lib under src floder.
2) Crate java class with name Excellib.java and write the code as shown below.

```
package com.lib;

import java.io.FileInputStream;

import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;

publicclass Excellib {

	publicstaticint getRowCount(String xlpath,String sheetName)
	{
		try
		{
			FileInputStream fos = new FileInputStream(xlpath);
			Workbook wb = WorkbookFactory.create(fos);
			int rowcount = wb.getSheet(sheetName).getLastRowNum();
			return rowcount;
		}
		catch(Exception e)
		{
			return -1;
		}

	}

	publicstaticstring getCellValue(String xlpath,String sheetName, int rownum,
int cellnum)
	{
		try
		{
			FileInputStream fos = new FileInputStream(xlpath);
			Workbook wb = WorkbookFactory.create(fos);
			String strvalue =
wb.getSheet(sheetName).getRow(rownum).getCell(cellnum).getStringCellValue();
			return strvalue;

		}
		catch(Exception e)
		{
```

```
                        return"";
                }
        }
}
```