

In eclipse

Create a java project in that project create Automation

Inside automation creates two empty folders 1. Jar 2. Driver

Check chrome version-- 92....

Selenium download: <https://www.selenium.dev/downloads/>

Download latest version 3.141.59

In the same page Go to browser click on chrome documents :

<https://chromedriver.chromium.org/> in this page under All versions Available download the

Respective chrome version 92....

After downloading extract the chromedriver file

Now copy the chrome extracted and paste in driver file present in Eclipse

Noe copy the jar file and paste in the jar file created in eclipse.

\*\*After that add only the jar file to build path...this will create a folder Referenced library

**{don't add exe file to build path else we will not be able to execute any thing}**

Create package qsp under this package create a class Demo

- Create an object of chrome driver class  
ChromeDriver c = new ChromeDriver(); [run -- will get an exception error  
now we need to set the path of chrome]
- System.setProperty("webdriver.chrome.driver","path");
- How to see path ..right click on chrome.exe file ...properties...copy the location and paste in the path .
- Run now we will get an empty chrome page.
- For generic/relative paths give path as “ ./driver/chromedriver.exe” (. dot means go to the current java project) \*\*recommended \*\*  
Since we can use the code in any system, if we use the environment it will work only in one system i.e, the system for which path is given.

Why do we add jar files to build paths?

Since this file is present in diff projects or files, we need to add it to the build path to get access for eclipse(since eclipse doest know where the file is present ) and we import because the compiler doest know where it is .

Automation : reducing the no. of resources.

Why ?

1. faster
2. Reduce man power
3. Efficiency / accuracy

### **Assignment :**

#### **1.what is automation?**

The use of software to create repeatable instructions and processes to replace or reduce human interaction with IT systems.

#### **2. What are the advantages of automation?**

- 1.Faster feedback cycle
- 2.reduce business expenses
- 3.higher test coverage
- 4.reusability of test suite
- 5.improved accuracy
- 6.quickly determine the stability of the build
- 7.eliminates human error
- 8.earlier detection of error
- 9.data driven testing
- 10.maximize ROI

Other than selenium - QTP/UFT , RANOREX, RFT, TEST COMPLETE

Selenium supports 14 diff coding languages ( java, python, js,C#, php, pearl,.....)

Platform independent

Selenium is a community in which we will learn selenium webdriver..

Using selenium we can automate only web applications. (only functional testing)

For performance testing we need to integrate it with the LOADRUNNER & J meter.

Client server app -->Selendroid , APPIUM.

Stand alone app → Winium

## **INTRODUCTION OF SELENIUM**

Selenium is a free and open source, web application automation tool ,or functional testing web application automation tool.

### **1. From where we can download the selenium s/w for free?**

<https://www.selenium.dev/downloads/>

### **2. What is the Latest stable version of selenium - 3.141.59**

### **3. What are the flavors of selenium ?**

Selenium core, Selenium RC(remote control), selenium IDE(record and play). Selenium webdriver, Selendroid, appium, Winium, Selenium grid.

### **4. What are the browsers supported by selenium?**

Google chrome, mozilla firefox, internet explorer, microsoft edge, opera, saffari, html unit, phantom js.

### **5. What OS is supported by selenium?**

Windows, linux, IOS and other flavors of UNIX like android, mac, ubuntu etc.

### **6. What are the languages supported by selenium?**

java, python, javascript, C#, php, pearl, objective C, Ruby, NodeJS, TCL, haskell, R, Elixir, Dart

### **7.Using selenium what type of app can we automate?**

Only Web application

### **8. Which OS is not supported by selenium?**

UNIX

### **9. Do we automate -ve test cases in selenium? Yes**

### **10. Can we do performance testing in selenium?**

No but we can integrate selenium with J meter

### **11. What type of test cases do we automate?**

We can automate functional test cases, integration, system, smoke test cases etc.

But it should be a part of the regression test suite.

### **12. Which type of test cases do we automate first?**

Smoke test cases (sanity, dry run, build verification testing)

### **13. Is 100% automation possible? NO**

#### 14. Why is it not possible?

Because we don't have technology to automate the features or it may be very costly or it may require manual intervention. Example: card swiping, biometric scanning, captcha, OTP, game testing, barcode scanning, verification of audio and video clips, etc.

#### 15. which OS and browser is supported by selenium IDE?

Windows and firefox

=====

Selenium is developed using JAVA.

To use selenium in other languages we need to use Language Binding / Client bindings.

ChromeDriver acts as a translator for Google chrome

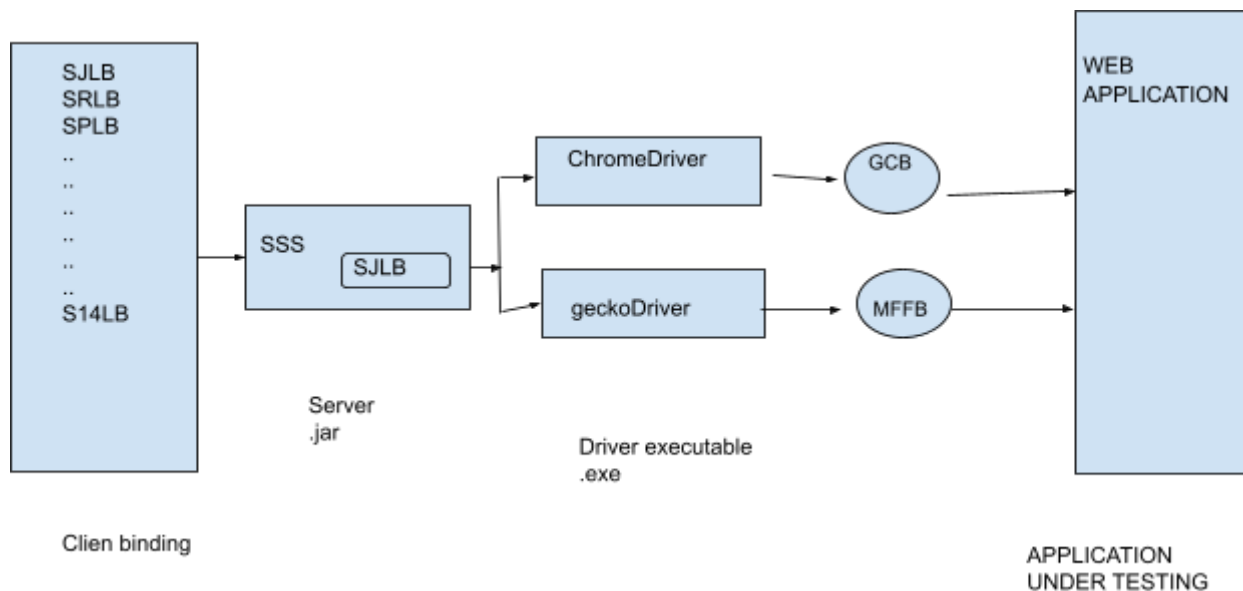
For mozilla firefox - geckoDriver

With the help of a web executable we are able to open Web Applications.

Selenium uses JSON wire protocol while sending and getting responses.

=====

#### Architecture of SELENIUM



Selenium supports 14 diff. Coding languages such as java, python, ruby etc... These are called Language binding or Client binding.

This client binding communicates with selenium server standalone. Then performs action of the browser with the help of driver executables, in order to perform action it uses JSON wire protocol (javascript object notation).

Selenium server internally contains selenium java client binding also.hence by installing selenium we use only jar file and driver executable files.

### **Steps to install selenium:**

1. Download selenium server jar file and browser specific driver executable file in the download page of selenium website .
2. Extract the driver executable file (unzip the file)
3. in eclipse create a new java project and create two new folders such as jar and driver inside the java project.
4. Copy and paste the driver executable file into the driver folder.
5. Similarly copy and paste selenium jar file into jar folder.
6. Add jar file to build path.so that we can use import statement to import required classes of selenium.

**Note:** do not add driver.exe file to the build path, because eclipse will not allow us to add exe file to the build path.

7. To specify the path either we add it manually to the environment variable or we should add it programmatically using system.setProperty.

8. While specifying the path we use relative path by specifying dot . operator at the beginning which represents the current java project.

9. We should set the path before opening the browser or else we get an illegal state exception error.

10. For easy maintenance we set the path of all drive executables in the static block as shown below

```
package qsp;
```

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

```
public class Demo {
```

```

    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver",
            "./driver/chromedriver.exe");
        ChromeDriver c= new ChromeDriver();
        c.get("https://www.google.com"); // open
        String title = c.getTitle(); // google
        System.out.println(title);
        //Thread.sleep(4000); // slow the closing process
        c.close(); // close
    }
}

```

**Which protocol is used to communicate with browsers?** JSON wire protocol.  
**How selenium performs action on the browser?** By calling native methods of the browser.

### **Architecture of Selenium WebDriver (or) Selenium java language binding**

Searchcontext I

WebDriver I

JavaScriptExecutor I

TakeScreenshot I

Remotewebdriver C

- Search context is the super most interface which is extended by webdriver interface. Webdriver interface is implemented as remotewebdriver class.
- Remotewebdriver class implements other interfaces also such as javascriptexecutor, take screenshot etc.
- All the browser specific classes such as chromedriver, firefoxdriver, internetexplorerdriver etc. extends remotewebdriver class.

**Note:** Eclipse will display methods of object class also when we press . dot after any reference variable. To hide it go to the window tab click on preferences >> java >> appearances >> type filters >> click on add >> type: java.lang.Object >> click on OK >> apply and close.

### **Methods of search context:**

findElement(By args):WebElement

findElements((By args): List<WebElement>

### **Methods of WebDriver**

close()

get()

getCurrentUrl()

getPageSource()

getTitle()

getWindowHandle()

getWindowHandles()

manage()

navigate()

quit()

switchTo()

### **Methods of JavaScriptExecutor**

executeScript()

executeAsyncScript()

### **Methods of TakeScreenshots**

getScreenshotAs()

### **Methods of WebElement**

clear()  
click()  
getAttribute()  
getCssValue()  
getLocation()  
getRect()  
getSize()  
getTagName()  
getText()  
isDisplayed()  
isEnabled()  
isSelected()  
sendKeys()  
submit()

**Q: How do you enter a URL without using get() ?**

By using navigate().to()

**Q: What is the difference b/w get() and navigate() ?**

By using get() we can enter the URL only, whereas by using navigate we can enter the URL, click on back, forward and refresh also.

**Q: What is the difference b/w get() and to() ?**

To () internally calling get() hence there is no diff b/w get() and to(), both are used to enter the URL.

**Q: What is upcasting, it is used in selenium and why ?**

Converting a sub class object into a super type is called upcasting.

Ex. WebDriver driver = new ChromeDriver();

In selenium we use upcasting so that we can execute the same script on any browser.

**Q: Explain WebDriver driver = new ChromeDriver()**

WebDriver :is an interface

driver :is a reference variable

= :is an assignment operator

new :is a keyword used to create an object

ChromeDriver() :is a constructor



; is used to end the statement (statement delimiter)

**Q: How do you close the browser without using close() ?**

By using quit().

**Q: Difference b/w close() and quit() ?**

close() closes the current browser whereas quit() closes all the browsers.

**Q: How do you delete all the cookies present in the browser?**

By using - driver.manage().deleteAllCookies();

**Q: How do you maximize the browser?**

By using - driver.manage().window().maximize();

**Q: Write a script to print HTML code of google.com ?**

By using - get.PageSource();

## **WebElement**

-Anything present on the web page is called as webelement such as button, text box, link, etc.

-Webelements are created using HTML language(hypertext markup language).

-HTML element contains 3 things

- 1.Tag
- 2.Attribute
- 3.Text

Example. HTML code of a login button is <div id="d1">Login</div>

In the above HTML code div is the Tag, id is the Attribute name, d1 is the attribute value, Login is the Text (visible text).

- In order to see the HTML code of the required element right click on that element and select the option Inspect. If right clicking is disabled on the page then press Ctr+shift+i or F12. It will display a developer toolbar.
- In order to inspect another element click on the inspect button and then click on the required element.

In Selenium before performing any action such as clicking, typing, selecting,etc We should find the element 1st using locators.

## **Locators**

Locators are used to find the element, in selenium there are 8 types of locators and all of them are static methods present in **By class**.

By is an abstract class.

- 1.tagName()
- 2.id()
- 3.name()
- 4.className()
- 5.linkText()
- 6.partialLinkText()
- 7.cssSelector()
- 8.xpath()

-All the locators will take the String as an argument.

**note** : in order to create the static web page type the below code in notepad and save it as Demo.html on the desktop.

```
<html>
<body>
<a id="d1" name="n1" class="c1" href="https://qspiders.com/">Qspiders</a><br>
<a id="d1" name="n1" class="c1" href="https://jspiders.com/">Jspiders</a>
</body>
</html>
```

**Q: Write a program to click on the Qspiders /Jspiders link present in the static web page using tagName locator ?**

```
public class LocatorCls {
    static {
        System.setProperty("webdriver.chrome.driver",
            "./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        WebDriver driver = new ChromeDriver();
```

```

        driver.get("file:///C:/Users/SATEESH/Desktop/Demo.html");
        WebElement e = driver.findElement(By.tagName("a"));
        e.click();
    }
}

```

**Q: what is the return type of findElement() ?**

WebElement

**Q: what is the argument accepted by findElement() ?**

File type

**Q: If the specified locator is matching with multiple elements what findElements does ?**

It takes first matching element or it returns the address of the first matching Element.

**Q: If the specified locator is not matching with any of the elements then what findElement does ?**

It throws NoSuchElementException.

**Q: write a script to click on the google link by using tagname, id, name, and classname locators.**

Jspiders/ Qspiders

```

public class LocatorAssign {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/Demo.html");
        driver.findElement(By.tagName("a")).click();
    }
}

```

```

        driver.navigate().back();
        driver.findElement(By.id("d1")).click();
        driver.navigate().back();
        driver.findElement(By.name("n1")).click();
        driver.navigate().back();
        driver.findElement(By.className("c1")).click();
    }
}

```

### **linkText() and partialLinkText()**

- linkText() and partialLinkText() locators can be used to find the link. If we try it on any other type of element we get NoSuchElementException.

Example: driver.findElement(By.linkText("Google")).click()

- If the text of the link is changing partially then we can use partialLinkText();

Example: HTML code of a link is: <a.....>Inbox(7)</a>

Selenium code is: driver.findElement(By.partialLinkText("Inbox")).click();

**\*\* we can use linkText() and partialLinkText() only when the tag name is a \*\***

### **Limitations of partialLinkText()**

1.Element should be a link. Ex: <span>Inbox(7)</span>

2. Text should be partially changing. Ex: <span>25</span>

### **cssSelector()**

-Css stands for Cascading style sheet.

-cssSelector is one of the locator in selenium.

- the syntax is :

Tag[attribute name = 'attribute value']

Example: a[id='d1']

a[name='n1']

a[class='c1']

a[href='https://jspiders.com/']

These all are called css expressions.

- We can check the css expression in the browser by using following steps:

1. Inspect any element.
2. Press Ctrl+f .

3. Type the above expression.
4. 1 of 1 >> 1 matching element
5. 1 of 3 >> multiple matching elements
6. 0 of 0 >> no matching elements

Example for cssSelector

```
public class cssSelectorCls {
    static {

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

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        WebDriver driver= new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/Demo.html");
        driver.findElement(By.cssSelector("a[id='d1']")).click();
        driver.navigate().back();
        driver.findElement(By.cssSelector("a[name='n1']")).click();
        driver.navigate().back();
        driver.findElement(By.cssSelector("a[class='c1']")).click();
        driver.navigate().back();

        driver.findElement(By.cssSelector("a[href='https://jspiders.com/']")).click();
        driver.navigate().back();
    }

}
```

**Note:**

Limitation of cssSelector is we can't use text or it doesn't support text.

**Note:** if we make any syntax mistake while writing css expression or xpath, then we get an InvalidSelectorException.

**xpath()**

-Path of the element in a HTML tree is called xpath. While writing xpath its starts with .(dot) which represents the current web page or html document.

-Using .(dot) is not mandatory.

-To navigate from parent element to child element we use / (single forward slash).

-xpath expression is .html/body/a (or) html/body/a

-selenium code for xpath is

```
driver.findElements(By.xpath("/html/body/a")).click
```

There are 2 types of xpath()

### 1. Absolute xpath

-Starting from the html tag to the desired element or the required element is called an absolute xpath.

-All the below xpath are called absolute xpath.

### Xpath index

In xpath we can use an index which starts from 1, then if there is another element under the same parent with the same tag then index becomes 2 and so on.

```
Html
|__ body
    |
    1 |__ div
        1 |__ a link1
        2 |__ a link2
    |
    2 |__ div
        1 |__ a link3
        1 |__ img image
        2 |__ a link4
```

Link 1: /html/body/div[1]/a[1]

Link 2: /html/body/div[1]/a[2]

Link 3: /html/body/div[2]/a[1]

Link 4: /html/body/div[2]/a[2]

```
{here it will check in both div}
```

### F. xpath by group index

// represents descendant

Image: //div/img (or) //img (or) //div/img[1]

**Q: How do you find all the links and all the images for any given web app or web page?**

`//a | //img`

Link1 and link 4 : `//div[1]/a[1] | //div[2]/a[2]`

Link 2 and link3 : `//div[1]/a[2] | div[2]/a[1]`

**Q: What is the difference b/w `//a` and `//table//a` ?**

`//a` matches with all the links present anywhere in the web page.

`//table//a` matches with all the links which are inside the table.

P	Q
A	B
C	D

Html

```
|__body
  |__a P
  |__a Q
  |__table
    |__a A
    |__a B
    |__a C
    |__a D
```

`//a >>> PQABCD`

`//table//a >>> ABCD`



### **A. xpath by Attribute:**

```
//tag[@AN ='AV']
```

We can also use attribute of an element while using xpath.

The syntax is

1. //tag[@AttributeName = 'AttributeValue']
2. //tag[@AN = 'AV' and @AN= 'AV']
3. //tag[@AN ='AV' or @AN ='AV']
4. //tag[not(@AN ='AV')]

Html code of the static page is

**Q: write an xpath to identify the following elements**

**1. all the text box**

```
>> //input[@type='text']
```

**2.all the buttons**

```
>> //input[@type='button']
```

**3.all the checkbox**

```
>> //input[@type='checkbox']
```

**4. All the text box and all the buttons**

```
>> //input[@type='text' or @type='button']
```

**5.Only first text box**

```
>> //input[@type='text' and @value='A']
```

**6.Only second text box**

```
>> //input[@type='text' and @value='B']
```

**7. 1st text box and 1st button**

```
>> //input[@value='A' ]
```

### **8.Only selected checkbox**

```
>> //input[@checked] or //input[@type='checkbox' and @checked]
```

### **9.Unselected checkbox**

```
>> //input[not(@checked) and @type='checkbox']
```

## **B. xpath by text function**

In xpath we can also use text function

The syntax is: //tag[text() = 'text value'] or //div[. = 'text value'] dot represents text()

Html code of login button is

Ex1- <div>Login</div>

Xpath is: //div[text() = 'Login']

Ex2-

<td id="headerContainer" class="header">Please identify yourself</td>

Xpath is: //td[text()='Please identify yourself']

## **C. xpath by contains function**

\_\_\_\_\_ If the text value is changing partially then we can use the contains function on the xpath.

The syntax is: //tag[contains(text(),value)]

Html code is: <nobr>actiTIME 2020 Online</nobr>

Xpath is: //nobr[contains(text(),'actiTIME')]

## **Q: How do you identify the element without using partialLinkText ?**

By using xpath by contains function.

Html code: <a>Index(6)</a>

Xpath: //a[contains(text(),'Inbox')]

\*We can use the contains function for attributes also.

The syntax is: //tag[contains(@attribute name,'attribute value')]

Html code:

```

```

Xpath: //img[contains(@class,'logo')]

## Assignment

**Write a script to login to actiTIME application.**

Use SendKeys('admin')

### **D. traversing xpath**

Navigating from one element to another element is called traversing.

- There are 2 types of traversing
  1. Forward traversing
  2. Backward traversing
- Navigating from parent element to child element is called forward traversing.
- Navigating from child element to parent element is called backward traversing.
- Sample html code is

```
<table border ="1">
<tr>
<td>java</td>
<td>100</td>
</tr>
<tr>
<td>selenium</td>
<td>300</td>
</tr>
</table>
```

java	100
selenium	300

```

Html
|__body
    |__table
        |__tbody
            1 |__tr
                1 |__td java
                2 |__td 100
            2 |__tr
                1 |__td selenium
                2 |__td 300

```

Example for forward traversing:(html to selenium)

/html/body/table/tbody/tr[2]/td[1]

Or

//td[text() ='selenium'] or //td[. = 'selenium']

Example for backward traversing:(from selenium to html)

//td[.='selenium']/../..../..

### **E. independent dependent xpath**

If the value is changing completely then we handle it using independent dependent xpath.(to handle dynamic web table also)

-Here we write the xpath using backward and forward traversing.

-We always start from an independent element and end with a dependent element.

```

          tr
        1_____|____2
        |      |
        td      td
    Selenium    300

```

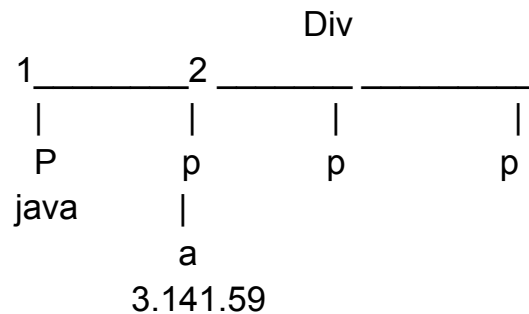
Xpath: //td[.='selenium']/../td[2]

### **Steps to construct HTML tree and write xpath:**

1. Identify the independent and dependent element.
2. Inspect the independent element and write the xpath for it.

3. Move the mouse pointer in upward direction step by step till it highlights both independent and dependent elements. It will be the common parent .
4. Add its path by using ../ Above the code of the independent element.
5. Use the down arrow key to navigate from common parent to dependent element.
6. Add its path below the common parent.

**Q: Write an xpath to identify stable version of java present under language bindings?**



**Q: Write an xpath to identify API docs link for Ruby ?**

`//p[.='Ruby']/../p[5]/a`

**Q: Write an xpath to identify the release candidate of python ?**

`//p[.='Python']/../p[3]/a`

**Q: Write an xpath to identify the status of testing type of work present in the type of work web table under settings icon of actiTIME ?**

`//a[.='testing']/../../td[2]`

**Q: Write an xpath to identify the set by default link for manufacturing type of work?**

`//td[@class='listtblcell billingTypeCell']/../td[5]/span`

**Q: Write an xpath to identify the rate for engineering type of work?**

`//td[@class='listtblcell billingTypeCell']/../td[4]/span`

**Q: Write an xpath to identify the price of iphone 12 pro present in amazon.in ?**

```
//span[.='New Apple iPhone 12 Pro (128GB) - Pacific  
Blue']/../../div[3]/div[1]/div//div[2]/a/span/span
```

**Q: Write an xpath to identify the price of mi band 5 present in flipkart ?**

```
(//div[.='Mi Smart Band 5'])[1]/../../div//div[3]/div/div/div/div
```

**Q: Write an xpath to identify the price of hrx by hrithik roshan t shirts present under men's section of myntra?**

```
(//h3[.='HRX by Hrithik Roshan'])[1]/../div/span[1]/span[1]
```

**Q: Write an xpath to identify the price of libas kurtas present under the women's section of myntra?**

```
(//h3[.='Libas'])[1]/../div/span/span
```

**Q: Write an xpath to identify the price of h&m jeans present under the kids section of myntra?**

```
(//h3[.='H&M'])[1]/../div/span/span
```

## **F. xpath by grouping index**

While writing the xpath expression we can write the expression within the () and we can specify the index outside the () which is called a group index.

Html

```
  |__body
    1 |__div
      1 |__a A
      2 |__a B
    2 |__div
      1 |__a C
      2 |__a D
```

xpath : (//a)[1] >>> A

A

B
C
D

When we use group index 1s it will execute the expression present inside the (), in this example 1st it will execute //a and it will store matching elements i.e, ABCD links into xpath array where index starts from 1.

Then it will identify the matching element based on the index which is specified outside the ( ), in this example it matches with 1st link A.

//a >> ABCD            matches with all the link  
 (//a)[1] >> A           matches with the  
 (//a)[2] >> B  
 (//a)[3] >> C  
 (//a)[4] >> D  
 (//a)[last()] >> D  
 //a[1] >> A C           matches with all the link having index 1  
 (//a[1])[1] >> A  
 (//a[1])[2] >> C  
 //a[2] >> B D  
 (//a[2])[1] >> B  
 (//a[2])[2] >> D  
 (//a)[1] | (//a0[last()]) >> AD           matches with 1st and last link.

selenium	100
selenium	300

Xpath: (//td[.='selenium'])[2]/../td[2]    (selenium 300)

**Q: What are the frequently used locators in selenium ?**

Id, name, linkText or xpath

**Q: Write a script to login to actiTIME application ?**

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class actiTime {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://demo.actitime.com/login.do");
        driver.findElement(By.xpath("//input[@type
='text']")).sendKeys("admin");

        driver.findElement(By.xpath("//input[@type='password']")).sendKeys("manager");
        driver.findElement(By.xpath("//div[text() ='Login ' ]")).click();
    }
}
```

**Q: Write a script to remove the text from the text box in OpenSourceBilling app(email and password text box) ?**

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class RemoveTextValue {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        WebDriver driver =new ChromeDriver();
        driver.get("http://demo.opensourcebilling.org/en/users/sign_in");
        driver.findElement(By.id("email")).clear();
        driver.findElement(By.id("password")).clear();
    }
}
```



```
}  
}
```

**Q: Write a script to check whether the facebook logo is displayed or not ?**

```
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
  
public class VerifyLogo {  
    static {  
        System.setProperty("webdriver.chrome.driver",  
"./driver/chromedriver.exe");  
    }  
    public static void main(String[] args) {  
        WebDriver driver =new ChromeDriver();  
        driver.get("https://www.facebook.com/");  
        boolean logo =  
driver.findElement(By.xpath("//img[@alt='Facebook']")).isDisplayed();  
        if(logo==true)  
            System.out.println("Logo is displayed");  
        else  
        {  
            System.out.println("Logo is not displayed");  
        }  
        driver.close();  
    }  
}
```

### **Assignment**

**Write a script to check whether the login button is working or not in facebook.com**

```
isEnabled()  
  
public class loginButton {  
    static {
```

```

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

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://en-gb.facebook.com/login/");
        boolean result = driver.findElement(By.id("loginbutton")).isEnabled();
        System.out.println(result);
        driver.close();
    }
}

```

**Write a script to print the status of the checkbox present in actiTIME app ?**  
 isSelected()

```

public class ckeckBox {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://demo.actitime.com/login.do");
        boolean result =
driver.findElement(By.xpath("//input[@name='remember']")).isSelected();
        System.out.println(result);
        driver.close();
    }
}

```

**Write a script to print or get the text of a forgot your password link present in actiTIME app?**

```
getText()

public class PrintText {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://demo.actitime.com/login.do");
        driver.manage().window().maximize();
        String text =
driver.findElement(By.id("toPasswordRecoveryPageLink")).getText();
        System.out.println(text);
        driver.close();
    }
}
```

**Write a program to click on the submit button present in OpenSourceBilling app without using click() ?**

submit()

**Note:**We can use submit() only when type= submit attribute is present

```
public class submitButton {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("http://demo.opensourcebilling.org/en/users/sign_in");
        driver.findElement(By.id("user_login_btn")).submit();
    }
}
```

```
}
```

**Q: Write a script to print the height and width of the login button of facebook ?**

```
public class logoHW {
    static {
        System.setProperty("webdriver.chrome.driver",
            "./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.facebook.com/");
        WebElement loginbtn = driver.findElement(By.name("login"));
        //int height =
        driver.findElement(By.name("login")).getSize().getHeight();
        //int width =
        driver.findElement(By.name("login")).getSize().getWidth();
        int height = loginbtn.getSize().getHeight();
        int width = loginbtn.getSize().getWidth();
        System.out.println("height: "+height);
        System.out.println("width: "+width);
        driver.close();
    }
}
```

### **Assignment**

**Write a script to check whether height and width of username and password text box are equal or not for actiTIME ?**

```
public class actiTimeHW {
    static {
        System.setProperty("webdriver.chrome.driver",
            "./driver/chromedriver.exe");
    }
}
```

```

public static void main(String[] args) {
    WebDriver driver = new ChromeDriver();
    driver.get("https://demo.actitime.com/login.do");
    WebElement us = driver.findElement(By.name("username"));
    int h1 = us.getSize().getHeight();
    int w1 = us.getSize().getWidth();
    WebElement pwd = driver.findElement(By.name("pwd"));
    int h2 = us.getSize().getHeight();
    int w2 = us.getSize().getWidth();
    if((h1==h2)&&(w1==w2))
        System.out.println("height and width of username and
password is same");
    else
        System.out.println("height and width of username and
password is different");
    driver.close();
}
}

```

**Write a script to print a text of submit button along with the tag name and class attribute of OpenSourceBilling app ?**

getTagName() , getText(), getAttribute(class)

```

public class getTxtTagCls {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("http://demo.opensourcebilling.org/en/users/sign_in");
        WebElement result =
driver.findElement(By.xpath("//button[@name='btn_login']"));
        String s = result.getTagName();
    }
}

```

```

        String s1 = result.getAttribute("class");
        String s2 = result.getText();
        System.out.println(s);
        System.out.println(s1);
        System.out.println(s2);
        driver.close();
    }
}

```

**Q: write a script to check whether username and password text box are properly aligned or not?**

```

public class VerifyAlign {
    static {
        System.setProperty("webdriver.chrome.driver",
            "./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://demo.actitime.com/login.do");
        int
x1=driver.findElement(By.name("username")).getLocation().getX();
        int x2=driver.findElement(By.name("pwd")).getLocation().getX();
        if(x1==x2)
            System.out.println("username and password are aligned
properly");
        else
            System.out.println("username and password are not aligned
properly");
        driver.close();
    }
}

```

**Q: write a script to check whether gender radio buttons are properly aligned or not in facebook.com after clicking the create new account button ?**

Use Thread.sleep(2000) after clicking

```
public class radioAlign {
    static {
        System.setProperty("webdriver.chrome.driver",
            "./driver/chromedriver.exe");
    }

    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.get("https://en-gb.facebook.com/");
        driver.findElement(By.xpath("(//a[@role='button'])[2]")).click();
        Thread.sleep(2000);
        int x1 =
            driver.findElement(By.xpath("(//input[@type='radio'])[1]")).getLocation().getY();
        int x2 =
            driver.findElement(By.xpath("(//input[@type='radio'])[2]")).getLocation().getY();
        int x3 =
            driver.findElement(By.xpath("(//input[@type='radio'])[3]")).getLocation().getY();
        if(x1==x2 && x1==x3)
            System.out.println("radio buttons are aligned properly");
        else
            System.out.println("radio buttons are not aligned properly");
        driver.close();
    }
}
```

**Q: write a script to print the color of the forgotten password link present in facebook.com?**

getcssValue() is used to fetch the value of css like font, color, etc.

```
public class PrintColor {
```

```

        static {
            System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
        }

        public static void main(String[] args) {
            WebDriver driver = new ChromeDriver();
            driver.get("https://en-gb.facebook.com/");
            WebElement link = driver.findElement(By.linkText("Forgotten
password?"));
            String color = link.getCssValue("color");
            System.out.println("color: "+color);
            String fontType = link.getCssValue("font-family");
            System.out.println("fontType: "+fontType);
            String fontSize = link.getCssValue("font-size");
            System.out.println("fontSize: "+fontSize);
        }
    }
}

```

## **findElements()**

- In order to handle multiple elements we go for findElements() .
- return type of findElements is list of WebElement .
- list should be imported from the java.util package.
- If the locators matching with multiple elements it will return the address of all the Matching elements.
- if the locators are not matching with any of the elements then findElements() returns an empty list.

.....image.....

Html code:



```

<html>
<body>
<a id="d1" name="n1" class="c1" href="https://qspiders.com/">Qspiders</a><br>
<a id="d1" name="n1" class="c1" href="https://jspiders.com/">Jspiders</a>
</body>
</html>

```

**\*Q: Write a script to print all the links present in Amazon.in ?**

```

public class HandlingMultiElmt {
static {
    System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
}

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.amazon.in/");
        List<WebElement> allLinks = driver.findElements(By.tagName("a"));
        int count = allLinks.size();
        System.out.println(count);
        for(int i=0;i<count;i++) {
            WebElement link = allLinks.get(i);
            String text = link.getText();
            System.out.println(text);
        }
        driver.close();
    }
}

```

**Note:**

In order to take the input from the user use the below code-

```

public class HandlingMultiElmt {
static {
    System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
}

```

```

public static void main(String[] args) {
    System.out.println("enter the url");
    Scanner sc = new Scanner(System.in);
    String url = sc.nextLine();
    WebDriver driver = new ChromeDriver();
    driver.get(url);
    List<WebElement> allLinks = driver.findElements(By.tagName("a"));
    int count = allLinks.size();
    System.out.println(count);
    for(int i=0;i<count;i++) {
        WebElement link = allLinks.get(i);
        String text = link.getText();
        System.out.println(text);
    }
    driver.close();
}
}

```

### **Automate the following scenario**

- 1.open the chrome browser**
- 2.go to google.com type java in the search text box.**
- 3.find all the auto suggestions and print the count of auto suggestions**
- 4.print the text of all auto suggestions**
- 5.select the first auto suggestion.**

```

public static void main(String[] args) throws InterruptedException {
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.google.com/");
    // enter java into the search text box
    driver.findElement(By.name("q")).sendKeys("java");
    Thread.sleep(2000);
    // store all the links inside the list matching with the word java
}

```

```

        List<WebElement> autosug =
driver.findElement(By.xpath("//span[contains(.,'java')]"));
        int count = autosug.size();
        System.out.println(count);
        for(int i=0;i<count;i++)
        {
            //WebElement allsugg = autosug.get(i);
            String text = autosug.get(i).getText();
            System.out.println(text);
        }
        // using for each loop
        /*for(WebElement sugg:autosug) {
            System.out.println(sugg.getText());
        }
        */
        //click on the 1st suggestion
        autosug.get(0).click();
        driver.close();
    }
}

```

### Assignment:

**Q:Write the above 2 programs using for each loop**

**Q:Automate the following scenario**

- 1.open the browser.
2. Go to flipkart.com type iphone in the search text box find all the auto suggestions.
- 3.print all the auto suggestions along with count(using for each loop).
- 4.click on the last auto suggestion.

**\*Q:Difference b/w findElement() and findElements()**

findElement()	findElements()
---------------	----------------

1.Return type of findElement is WebElement. 2.if the locator is not matching it will throw a NoSuchElementException. 3. If the locators are matching with multiple elements it will return the 1st matching element. 4. Used to handle a single element.	1.Return type is list of WebElements. 2. Here it returns an empty list. 3. It returns all the matching elements 4. Used to handle multiple elements.
---	---

## **Synchronization**

- Process of matching the selenium speed with the application is called synchronization.
- most of the time selenium is faster than the application because of this reason we may not get the expected result or we end up with NoSuchElementException.

**Example** for Thread.sleep( ):

```
public class actiTime {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }

    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.get("https://demo.actitime.com/login.do");
        driver.findElement(By.xpath("//input[@type
='text']")).sendKeys("admin");
        driver.findElement(By.xpath("//input[@type='password']")).sendKeys("manager");
        driver.findElement(By.xpath("//div[text() ='Login ' ]")).click();
        Thread.sleep(5000);
        driver.findElement(By.className("logout")).click();
    }
}
```

## **Implicit wait()**

-In selenium there are different ways to synchronize the sleep

-One of the frequently used options is Implicit wait.

The Syntax is -

```
driver.manage().timeouts().implicitlyWait( time, TimeUnit.SECONDS);
```

It takes 2 arguments

1.time duration(long)

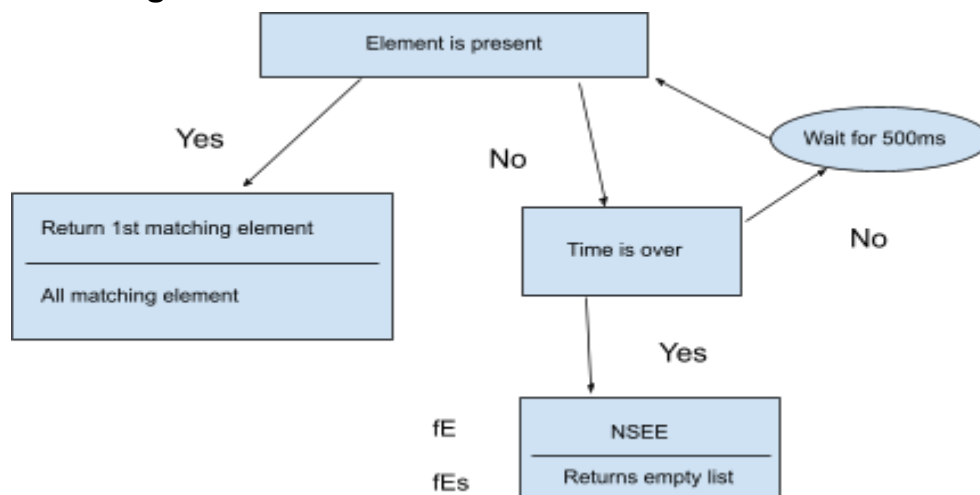
2.timeUnit (microseconds, milliseconds, seconds. Minutes, hours, days etc)

The specified duration is used only by findElements and findElement statement

- means implicit wait will work only for findElements() and findElement().

Default value for implicit wait is zero second.

### Flow diagram



When the control comes to any findElement or findElements statement it will check whether the element is present or not.

If the element is present then the findElement() will return the 1st matching element, whereas findElements returns all the matching elements.

If the specified element is not present then it will check for the timeout.

If the time is over then findElement() throws NoSuchElementException, whereas findElements() returns empty list.

If the time is not over it will wait for 500ms(half second) which is called the polling period.

Then it will continue to check whether an element is present or not.

```
public class Synchronization {  
    static {
```

```

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

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(10,
TimeUnit.SECONDS);
        driver.get("https://demo.actitime.com/login.do");
        driver.findElement(By.xpath("//input[@type
='text']")).sendKeys("admin");

        driver.findElement(By.xpath("//input[@type='password']")).sendKeys("manager");
        driver.findElement(By.xpath("//div[text() ='Login '])).click();
        driver.findElement(By.className("logout")).click();
    }
}

```

**Q:Can we specify implicit wait statements multiple times in the script?**

Yes

**Q: Is it necessary to write implicit wait statements multiple times?**

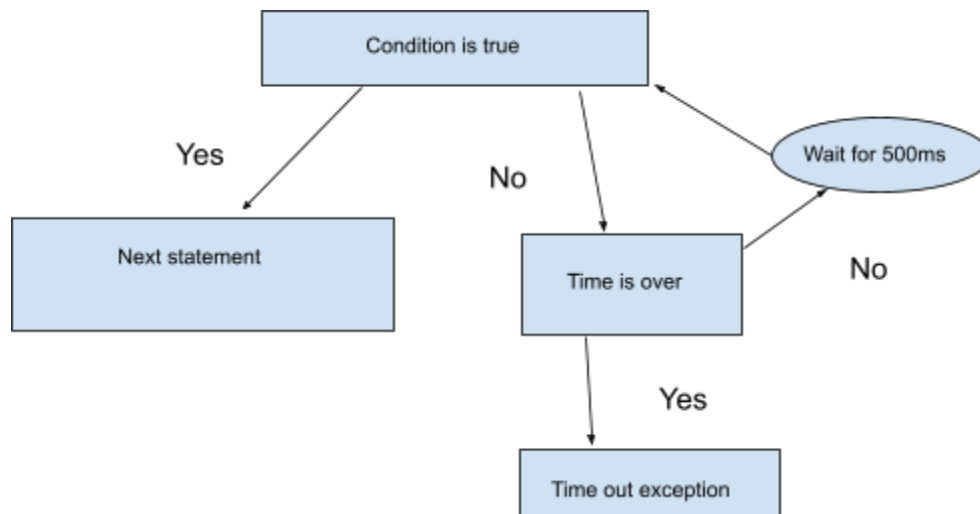
No

### **Explicit wait()**

In order to handle the synchronization of any method, we can use explicit wait. WebDriverWait itself is called an explicit wait, because we have to specify the waiting condition explicitly.

Syntax : `WebDriverWait wait = new WebDriverWait(driver,10);`  
`wait.until(ExpectedConditions.titleIs("actiTIME - Enter Time-Track"));`

Flow diagram



When the controls comes to `wait.until` statement it will check the specific condition.

If the condition is true it will go to the next statement, if the condition is false it will check for the time out.

If the time is over, it will throw `TimeoutException`, else it will wait for 500ms and it will continue to check the condition.

**Note:** in the above flow diagram titles is the condition.

By using explicit wait we can handle the synchronization of any statement but only one at a time.

Script:

```
public class ExplicitWait {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(10,
TimeUnit.SECONDS);
    }
}
```

```

        driver.get("https://demo.actitime.com/login.do");
        driver.findElement(By.xpath("//input[@type
='text']")).sendKeys("admin");

driver.findElement(By.xpath("//input[@type='password']")).sendKeys("manager");
        driver.findElement(By.xpath("//div[text() ='Login '])).click();
        WebDriverWait wait = new WebDriverWait(driver,10);
        wait.until(ExpectedConditions.titleContains("Enter"));
        // wait.until(ExpectedConditions.titleIs("actiTIME - Enter
Time-Track"));

        String title= driver.getTitle();
        System.out.println(title);
        driver.close();
    }
}

```

**Note:**

ExpectedCondition is an interface and ExpectedConditions is a class.

**Q: Can we handle the synchronization of findElement() using explicit wait?**

Yes.

```

public class expectedConditions {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://demo.actitime.com/login.do");
        driver.findElement(By.xpath("//input[@type
='text']")).sendKeys("admin");
    }
}

```



```

driver.findElement(By.xpath("//input[@type='password']")).sendKeys("manager");
    driver.findElement(By.xpath("//div[text() ='Login '])).click();
    WebDriverWait wait = new WebDriverWait(driver,10);

wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("logoutLink")));
    driver.findElement(By.id("logoutLink")).click();
    driver.close();
}
}

```

### **CustomWait:**

Handling the synchronization of the automation script by writing our own java code is called as custom wait.

Example

```

public class CustomWait {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://demo.actitime.com/login.do");
        driver.findElement(By.xpath("//input[@type
='text']")).sendKeys("admin");

driver.findElement(By.xpath("//input[@type='password']")).sendKeys("manager");
        driver.findElement(By.xpath("//div[text() ='Login '])).click();
        int i = 0;
        while(i<1000)
        {
            try
            {

```

```

        driver.findElement(By.id("logoutLink")).click();
        break;
    }
    catch(NoSuchElementException e)
    {
        i++ ;
    }
}
//driver.close();
}
}

```

**Q: What is the default value of the polling period?**

Half second or 500 ms

**Q: What are the different ways to sync the script ?**

ImplicitWait, ExplicitWait, Thread.sleep, CustomWait, fluentWait(it is mainly used to change the polling period and not widely used in industry)

**\*Q: What is the difference between Implicit and Explicit wait ?**

ImplicitWait	ExplicitWait
1.We don't specify the waiting condition explicitly.	1.We should specify the waiting condition explicitly.
2. We can handle the sync of all the findElement() and findElements().	2. We can handle the sync of any method but only one at a time.
3. After the duration we get NoSuchElementException.	3. After the duration we get a TimeoutException.

4. Time duration can be days, hours, minutes, seconds...etc.	4. Time duration will be only seconds.
--	--

**Q: write a script to check whether the login page is loaded within 15s or not?**

```
public class PageLoadTimeOut {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.manage().timeouts().pageLoadTimeout(15,
TimeUnit.SECONDS);
        try {
            driver.get("https://demo.actitime.com/login.do");
            System.out.println("the login page loading within 15s");
        }
        catch(Exception e) {
            System.out.println("the login page is not loading within 15s");
        }
        driver.close();
    }
}
```

**Q: Write a script to copy the text present in the email text box and paste it into the password text box for OpenSourceBilling app.**

```
public class copyPaste {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
}
```

```

public static void main(String[] args) {
    WebDriver driver =new ChromeDriver();
    driver.get("http://demo.opensourcebilling.org/en/users/sign_in");
    driver.findElement(By.id("email")).sendKeys(Keys.CONTROL+"ac");

    driver.findElement(By.id("password")).sendKeys(Keys.CONTROL+"av");
    }
}

```

### **Handling Listbox (dropdown or combo box)**

- There are two types of listbox
  - 1.single select listbox
  - 2.multi select listbox
- Whenever a listbox is created using select tag and content of the listbox is created using Option tag.
- Then to handle this listbox we use the select class of selenium. It should be imported from 'org.openqa.selenium.support.ui' package.
- Select class has a parameterised constructor where it takes an argument of type WebElement (address of the listbox).
- In order to perform action on the listbox we can use any one of the following methods:
  1. selectByIndex(int) >>>> takes int as an argument
  2. selectByValue(String) >>>> takes String as an argument
  3. selectByVisibleText(String)
  4. deSelectByIndex(int)
  5. deSelectByValue(String)
  6. deSelectByVisibleText(String)
  7. deSelectAll()
  8. getFirstSelectedOption()
  9. getAllSelectedOption()
  10. isMultiple()
  11. getOptions()
  12. getWrappedElement()

**Q: write a script to select the options present in the month listbox present in facebook?**

```
public class SelectClass {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        driver.get("https://en-gb.facebook.com/");
        driver.findElement(By.linkText("Create New Account")).click();
        WebElement monthlist = driver.findElement(By.id("month"));
        Select s= new Select(monthlist);
        s.selectByIndex(12);
        Thread.sleep(3000);
        s.selectByValue("3");
        Thread.sleep(3000);
        s.selectByVisibleText("Sep");
    }
}
```

### **Assignment**

**Q: Write a script to select your DOB on facebook.com?**

```
public class FacebookDob {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        driver.get("https://en-gb.facebook.com/");
        driver.findElement(By.linkText("Create New Account")).click();
        WebElement daylist = driver.findElement(By.name("birthday_day"));
```

```

        Select d = new Select(daylist);
        d.selectByIndex(14);
        WebElement monlist =
driver.findElement(By.name("birthday_month"));
        Select m = new Select(monlist);
        m.selectByValue("12");
        WebElement yearlist =
driver.findElement(By.name("birthday_year"));
        Select y = new Select(yearlist);
        y.selectByVisibleText("1996");
    }
}

```

### **Note:**

- If the specific option is duplicate then it will select the 1st matching element, if the specific option is not present then it will throw NoSuchElementException.
- if you try to use deSelect() on the single select listbox we get UnsupportedOperationException.
- While creating the object of the select class if we specify the WebElement which is not a listbox then we get UnexpectedTagNameException.

### **Handling multi-Select listbox**

HTML code to create static web page-

```

public class multiSelect {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/Hotel.html");
    }
}

```

```

        WebElement mtrlist = driver.findElement(By.id("mtr"));
        Select s = new Select(mtrlist);
        Thread.sleep(3000);
        s.selectByIndex(0);
        s.selectByValue("v");
        s.selectByVisibleText("dosa");
        Thread.sleep(2000);
        s.deselectAll();
        System.out.println("s.isMultiple()");
// this will check whether list box is multiple select or single select and returns
boolean value
        driver.close();
    }
}

```

**Q: Write a script to print the first selected element present in the SLV listbox?**

```

    public class FirstSelectedOpt {
        static {
            System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
        }

        public static void main(String[] args) {
            WebDriver driver =new ChromeDriver();
            driver.get("file:///C:/Users/SATEESH/Desktop/Hotel.html");
            WebElement slvList = driver.findElement(By.id("slv"));
            Select s = new Select(slvList);
            WebElement firstopt = s.getFirstSelectedOption();
            String text = firstopt.getText();
            System.out.println(text);
            driver.close();
        }
    }
}

```

**Q: Write a script to print all the selected elements present in the SLV listbox?**

```
public class AllSelected {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/Hotel.html");
        WebElement slvListbox = driver.findElement(By.id("slv"));
        Select s = new Select(slvListbox);
        List<WebElement> allSelectedOpt = s.getAllSelectedOptions();
        int count = allSelectedOpt.size();
        System.out.println(count);
        for(int i=0;i<count;i++)
        {
            String text=allSelectedOpt.get(i).getText();
            System.out.println(text);
        }
        driver.close();
    }
}
```

**Q: write a script to print all the options present in the MTR listbox?**

```
public class PrintAllOpt {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/Hotel.html");
        WebElement mtrListbox = driver.findElement(By.id("mtr"));
        Select s = new Select(mtrListbox);
        List<WebElement> allOpt = s.getOptions();
    }
}
```



```

        for(WebElement options:allOpt)
        {
            String text = options.getText();
            System.out.println(text);
        }
        driver.close();
    }
}

```

### Assignment:

1. Write a script to navigate to flipkart.com and search for iphone 12 and print all the iphones product name along with price in the below format ?

Product 1 -> price

Product 2 -> price

```

public class ProductPrice {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(5,
        TimeUnit.SECONDS);
        driver.get("https://www.flipkart.com/");

        driver.findElement(By.xpath("//input[@type='text']")).sendKeys("iphone
        12"+ Keys.ENTER);
        List<WebElement> proName =
        driver.findElements(By.xpath("//div[@class='_4rR01T']"));
        List<WebElement> price =
        driver.findElements(By.xpath("//div[@class='_30jeq3 _1_WHN1']"));
        int count = proName.size();
        System.out.println(count);
    }
}

```

```

for(int i=0;i<count;i++) {
    String text = proName.get(i).getText();
    String text1 = price.get(i).getText();
    System.out.println(text+ "--->" + text1);

}
driver.close();
}
}

```

**2. Write a script to search for wipro jobs in google.com and capture all the URLs (after navigating to search page by clicking enter)**

Inside for loop

allLinks = get(i).get(Attribute(href));

```

public class WiproUrl {
static {
    System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
}

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(5,
TimeUnit.SECONDS);
        driver.get("https://www.google.com/");
        driver.findElement(By.name("q")).sendKeys("wipro
jobs"+Keys.ENTER);
        List<WebElement> allLinks =
driver.findElements(By.xpath("//a[contains(.,'ipro')]"));
        int count = allLinks.size();
        System.out.println(count);
        for(WebElement allUrl:allLinks) {
            String url = allUrl.getAttribute("href");
            System.out.println(url);
        }
        driver.close();
    }
}

```

```

    }
}

```

**3. Write a script to navigate to bbc.com then capture 5 top latest business news and print it on the console.**

Use independent-dependent xpath

`//span[@class='top-list-item__bullet']/../h3`

```

public class BbcNews {
    static {
        System.setProperty("webdriver.chrome.driver",
            "./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver =new ChromeDriver();
        driver.get("https://www.bbc.com/");
        List<WebElement> topnews =
driver.findElements(By.xpath("//span[@class='top-list-item__bullet']/../h3"));
        int count = topnews.size();
        System.out.println(count);
        for(WebElement Lnews:topnews)
        {
            String text = Lnews.getText();
            System.out.println(text);
        }
    }
}

```

**4. Write a script to print all the options in mtr listbox without duplicates?**

Use hashset concept

```

public class NoDuplicateOpt {

```

```

        static {
            System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
        }
        public static void main(String[] args) {
            HashSet<String> hs = new HashSet<>();
            WebDriver driver = new ChromeDriver();
            driver.get("file:///C:/Users/SATEESH/Desktop/Hotel.html");
            WebElement mtrListbox = driver.findElement(By.id("mtr"));
            Select s = new Select(mtrListbox);
            List<WebElement> allOpt = s.getOptions();
            int count = allOpt.size();
            System.out.println(count);
            for(int i=0;i<count;i++) {
                String text= allOpt.get(i).getText();
                hs.add(text);
            }
            // this for each loop is to print the options vertically
            for(String text:hs) {
                System.out.println(text);
            }
            driver.close();
        }
    }
}

```

**5. Write a script to print all the duplicate options present in the MTR listbox?**

```

public class OnlyDuplicate {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        HashSet<String> hs = new HashSet<>();
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/Hotel.html");
    }
}

```

```

        WebElement mtrListbox = driver.findElement(By.id("mtr"));
        Select s = new Select(mtrListbox);
        List<WebElement> allOpt = s.getOptions();
        int count = allOpt.size();
        System.out.println(count);
        for(int i=0;i<count;i++) {
            String text= allOpt.get(i).getText();
            if(hs.add(text)==false){
                System.out.println(text);
            }
        }
        // here hashset will not allow duplicate value so the add() method will
        return false
    }
    driver.close();
}
}

```

## 6. WAS to print all the options present in MTR listbox in alphabetical order (with and without using TreeSet)?

Use Sort without TreeSet .

sorted:

```

public class AlphabeticalOrder {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        ArrayList<String> al = new ArrayList<String>();
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/Hotel.html");
        WebElement mtrlist = driver.findElement(By.id("mtr"));
        Select s = new Select(mtrlist);
        List<WebElement> allOpt = s.getOptions();
        for(int i =0;i<allOpt.size();i++) {

```

```

        String text = allOpt.get(i).getText();
        al.add(text);
    }
    Collections.sort(al);
    System.out.println(al);
    for(int i=0; i<al.size();i++) // to print vertically
    {
        System.out.println(al.get(i));
    }
    driver.close();
}
}

```

---

Using treeset:

```

public class mtrAlpa {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        TreeSet<String> ts = new TreeSet<String>();
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/Hotel.html");
        WebElement mtrListbox = driver.findElement(By.id("mtr"));
        Select s = new Select(mtrListbox);
        List<WebElement> allOpt = s.getOptions();
        int count = allOpt.size();
        System.out.println(count);
        for(int i=0;i<count;i++) {
            String text= allOpt.get(i).getText();
            ts.add(text);
        }
        for(String text:ts) {
            System.out.println(text);
        }
    }
}

```

```

    }
}
}

```

**{in hashSet we can't use normal for loop for ascending order since it doesn't preserve the insertion order and we can't use index value, whereas in arraylist we can use normal for loop }**

**\*Q: WaS to select all the options in the MTR listbox and deselect them in reverse order?**

```

public class DeselectInReverse {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/Hotel.html");
        WebElement mtrlist = driver.findElement(By.id("mtr"));
        Select s = new Select(mtrlist);
        List<WebElement> allOpt = s.getOptions();
        int count = allOpt.size();
        for(int i=0;i<count;i++) {
            s.selectByIndex(i);
            Thread.sleep(2000);
        }
        for(int i=count-1;i>=0;i--) {
            s.deselectByIndex(i);
            Thread.sleep(2000);
        }
        driver.close();
    }
}

```

}

## **Handlings Pop-ups**

- In selenium depending on the popup we write different types of code to perform action on the popup.
- Pop ups are generally categorised as follows
  1. Javascript or alert popup
  2. Hidden division or calendar popup
  3. File upload popup
  4. File download popup
  5. Notification popup
  6. Authentication popup
  7. Child window popup

## **JavaScript or Alert Popup (confirmation popup)**

### **Characteristics:**

1. We can not move this popup.
2. We can't inspect this popup.
3. This popup will have an OK button(alert) and if it contains an OK and CANCEL button(confirmation popup).
4. It will be present below the address bar in the middle section of the browser.

### **Solution:**

To handle javascript popup we use driver.switchTo().alert() statement

1. accept() : to click on the OK button.
2. dismiss() : to click on the CANCEL button.
3. getText() : to get the text present on the popup.
4. sendKeys() : to enter the text on the popup.

Example-



```

public class HandlingPopup {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();

driver.get("https://www.seleniumeasy.com/test/javascript-alert-box-demo.html");
        driver.findElement(By.xpath("//button[@class='btn
btn-default']")).click();
        WebDriverWait wait= new WebDriverWait(driver, 10);
        wait.until(ExpectedConditions.alertIsPresent());
        String text = driver.switchTo().alert().getText();
        driver.switchTo().alert().accept();
        /*[
        Alert a = driver.switchTo().alert();
            String text = a.getText();
            a.accept();
        // since we are using driver.switchTo().alert() multiples times, so to use easily we
        will store it in a variable
        ] */
        System.out.println(text);
        driver.close();
    }
}

```

All the above codes will work on confirmation popup also.

**Note:** if the popup is alert then there will be no difference between accept and dismiss, both will click on the OK button only.

## Hidden division or calendar popup

### Characteristics:

1. We can't move this popup.
2. We can inspect this popup.

### Solution:

1. We handle this popup by using findElement().

Example:

```
public class HiddenAlert {
static {
    System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
}

    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.flipkart.com/");
        Thread.sleep(3000);
        driver.findElement(By.xpath("(//button)[2]")).click();
        driver.close();
    }
}
```

## Assignment

### 1. Automate the following scenario

- a. Open the browser
- b. Enter the URL  
(<https://www.careinsurance.com/rhicl/proposalcp/renew/index-care>)
- c. Enter the policy no. as 123
- d. Click on DOB
- e. Select your DOB
- f. Enter you contact no. as 9845798457
- g. Click on lets renew button

```

public class CareInsuranceHiddenDivision {
static {
    System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
}

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("https://www.careinsurance.com/rhicl/proposalcp/renew/index-care");
        driver.findElement(By.className("form-control")).sendKeys("123");
        driver.findElement(By.id("dob")).click();
        WebElement mon =
driver.findElement(By.className("ui-datepicker-month"));
        Select s = new Select(mon);
        s.selectByValue("11");
        WebElement year =
driver.findElement(By.className("ui-datepicker-year"));
        Select s1 = new Select(year);
        s1.selectByValue("1996");
        driver.findElement(By.xpath("//a[.='14']")).click();

driver.findElement(By.id("alternative_number")).sendKeys("9854798547");
        driver.findElement(By.id("renew_policy_submit")).submit();

    }
}

```

## **File upload popup**

### Characteristics

1. We can move this popup.
2. We can't inspect this popup.
3. This popup will have an Open and cancel Button.
4. Title of the popup will be either Open or File upload.

Type this HTML code on the notepad and save it on the desktop.

```
<input type="file" id="cv"/>
```

Solution:

- a. To handle file upload popup we specify the absolute path of the file as an argument for sendKeys().
- b. sendKeys() should be used for the element on which if we click, which displays the popup.

Example:

```
public class FileUpload {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) throws InterruptedException {
        WebDriver driver= new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/FileUpload.html");
        Thread.sleep(4000);
        File f = new File("./data/Resume.docx");
        String absolutepath = f.getAbsolutePath();
        driver.findElement(By.id("cv")).sendKeys(absolutepath);
        driver.close();
    }
}
```

File is a class present in java.io package, it takes file path as an argument.

getAbsolutePath() is a non-static member present in file class, it converts relative path into absolute path.

### **\*Assignment**

**2. Write a script to login to naukri.com and upload your resume into it?**

```

public class NaukriResume {
static{
    System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
}

    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.naukri.com/");
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        driver.findElement(By.id("login_Layer")).click();
        driver.findElement(By.xpath("//input[@placeholder='Enter your active
Email ID / Username']")).sendKeys("hemavijayamadhavi220@gmail.com");

        driver.findElement(By.xpath("//input[@type='password']")).sendKeys("Vijaya30@"
);
        driver.findElement(By.xpath("//button[@type='submit']")).submit();
        driver.findElement(By.xpath("//div[text()='UPDATE
PROFILE']")).click();
        Thread.sleep(3000);
        File f = new File("./data/Resume.docx");
        String abspath = f.getAbsolutePath();
        driver.findElement(By.xpath("//input[@class='fileUpload
waves-effect waves-light btn-large']")[1]")).sendKeys(abspath);
    }
}

```

=====

[ Robot is a class  
keyPress is a non-static method present in robot class.  
KeyEvent is a static method having all the characters as constant.  
We can enter all the char by:  
robot r= new robot();  
keyPress( KeyEvent.VK\_Q);

Here VK stands for virtual keys.  
For other than numbers and alphabets keys we need to release the Key using keyRelease()  
sendKeys() works only for browsers. ]

=====

### **Robot class:**

Robot class is a java class present in java.awt package (abstract window toolkit)  
We use robot class whenever we want to perform any keyboard operations in windows.

In robot class we commonly use two methods

1. keyPress()
2. keyRelease()

Robot class works similar to sendKeys().

### **Q: Write a program to type qsp on the notepad?**

```
public class DemoRobotClass {  
  
    public static void main(String[] args) throws IOException, AWTException {  
        Runtime.getRuntime().exec("notepad");  
        Robot r = new Robot();  
        r.keyPress(KeyEvent.VK_SHIFT);  
        r.keyPress(KeyEvent.VK_Q);  
        r.keyRelease(KeyEvent.VK_SHIFT);  
        r.keyPress(KeyEvent.VK_S);  
        r.keyPress(KeyEvent.VK_P);  
    }  
}
```

### **Note:**

Runtime is a java class and it is a singleton class. Singleton class means it will allow you to create only one object at any instance of time.

### **File download popup:**

Characteristic with respect to firefox browser

1. We can move this popup.

2. We can't inspect this popup.
3. This popup will have an Open with and Save file radio button, along with Ok and Cancel button.

Solution:

1. By using Robot Class.

**Q: Automate the following scenario**

1. **Open the firefox browser.**
2. **Go to the download page of selenium.**
3. **Click on the download link of java.(which will display file download popup)**
4. **Select save file radio button by pressing Alt+s .**
5. **Click on the OK button by pressing enter.**

```
public class FileDownloadPopup {
    static {
        System.setProperty("webdriver.gecko.driver",
"./driver/geckdriver.exe");
    }
    public static void main(String[] args) throws AWTException,
InterruptedException {
        WebDriver driver = new FirefoxDriver();
        driver.get("https://www.selenium.dev/downloads/");
        driver.findElement(By.xpath("//p[.='Java']/../p[2]/a")).click();
        Thread.sleep(1000);
        Robot r = new Robot();
        r.keyPress(KeyEvent.VK_ALT);
        r.keyPress(KeyEvent.VK_S);
        r.keyRelease(KeyEvent.VK_ALT);
        r.keyPress(KeyEvent.VK_ENTER);
        r.keyRelease(KeyEvent.VK_ENTER);
    }
}
```

When we click on any type of download option in chrome browser, it will not display file download popup, instead of that it will start downloading file directly, hence we don't have to handle file download popup in chrome browser. In all other browsers we use Robot class.

### **Print popup**

Characteristics with respect to firefox

1. We can move this popup.(we can't move this popup in newer version)
2. We can't inspect this pop.
3. It will have a print and cancel button.

Solution

1. We handle this popup by using Robot class.

### **Note:**

We use Robot class to handle print popups in all the browsers except chrome. In the chrome browser we inspect the print button(elements) so we handle it by using findElement().

### **Assignment**

#### **1: Automate the following scenario**

1. Open the firefox browser.
2. Go to the download page of selenium.
3. Click on the link 3.141.59 under the latest stable version of selenium grid.
4. Click on the save file button on the popup.

#### **2: Write a script to download your resume from naukri.com by using the firefox browser.**

```
public class DownloadResume {  
    static {
```



```

        System.setProperty("webdriver.gecko.driver",
"./driver/geckodriver.exe");
    }
    public static void main(String[] args) throws InterruptedException,
AWTException {
        WebDriver driver = new FirefoxDriver();
        driver.get("https://www.naukri.com/");
        driver.manage().timeouts().implicitlyWait(60, TimeUnit.SECONDS);
        driver.findElement(By.id("login_Layer")).click();
        driver.findElement(By.xpath("//input[@placeholder='Enter your active
Email ID / Username']")).sendKeys("hemavijayamadhavi220@gmail.com");

driver.findElement(By.xpath("//input[@type='password']")).sendKeys("Vijaya30@"
);
        driver.findElement(By.xpath("//button[@type='submit']")).submit();
        driver.findElement(By.xpath("//div[text()='UPDATE
PROFILE']")).click();
        driver.findElement(By.xpath("//i[@title='Click here to download your
resume']")).click();
        Thread.sleep(2000);
        Robot r = new Robot();
        r.keyPress(KeyEvent.VK_ALT);
        r.keyPress(KeyEvent.VK_S);
        r.keyRelease(KeyEvent.VK_ALT);
        r.keyPress(KeyEvent.VK_ENTER);
        r.keyRelease(KeyEvent.VK_ENTER);

    }
}

```

### **3: Automate the following scenario**

- 1. Open the firefox browser.**
- 2. Go to goole.com.**
- 3. Press ctrl+p to get a print popup.**
- 4. Select the number of copies as 3 and click on the print button.**

## Notification popup

### Characteristics

1. We can't move this popup.
2. We can't inspect this popup.
3. This popup has an allow and block button.
4. It will be displayed below the address bar in the beginning.

### Solution

1. In order to handle this popup we change the settings of the browser, so that notification popup will not be displayed.
2. To change the settings of the browser we use addArguments() of the ChromeOptions class.

### Note

addArguments() is an example for method overloading.(where it takes string or list<string> as an argument).

For every browser we have respective Options classes like FirefoxOptions, internetExplorerOptions, OperaOptions etc.

In order to open the browser in modified settings we use parameterised constructor in each browser class.

new ChromeDriver() ----- opens the browser in default settings

new ChromeDriver(option) ----- opens the browser in modified settings.

The above statement is an example for constructor overloading.

```
public class NotificationPopup {
    static {
        System.setProperty("webdriver.chrome.driver",
            "./driver/chromedriver.exe");
    }

    public static void main(String[] args) {
        ChromeOptions option = new ChromeOptions();
        option.addArguments("--disable-notifications");
        WebDriver driver = new ChromeDriver(option);
        driver.get("https://www.yatra.com/");
    }
}
```

## Authentication Popup

### Characteristics

1. We can't move this popup.
2. We can't inspect this popup.
3. This popup will have a username and password text box along with a sign-in and cancel button.

### Solution

We handle this popup by sending username and password along with the URL inside the get().

### Test Data:

Username: admin

Password: admin

[https://the-internet.herokuapp.com/basic\\_auth](https://the-internet.herokuapp.com/basic_auth)

```
public class AuthenticationPopup {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();

driver.get("https://admin:admin@the-internet.herokuapp.com/basic_auth");
    }
}
```

## Child window popup or child browser popup

### Characteristics

1. We can move the popup.
2. We can inspect the popup.
3. This popup has minimize, maximize and close the button along with the address bar.

### Solution

1. To handle this popup we use getWindowHandle() , getWindowHandles() and driver.switchto().window() statement.

```
public class ChildWondowPopup {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com/");
        String wh=driver.getWindowHandle();
        System.out.println(wh);
        driver.close();
    }
}
```

### Note

- Address of the browser present on the desktop is called windowHandle (session id).
- In order to retrieve it we use getWindowHandle() .

**Q: Write a script to count the number of browsers opened in Naukri.com?**

```
public class NaukriChildWindows {
    static{
```

```

        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.naukri.com/");
        Set<String> allWh = driver.getWindowHandles();
        int count = allWh.size();
        System.out.println(count);
        driver.quit();
    }
}

```

**Q: write a script to print all the windowHandles present in the naukri .com?**

```

    public class NaukriChildWindows {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.naukri.com/");
        Set<String> allWh = driver.getWindowHandles();
        int count = allWh.size();
        System.out.println(count);
        for(String text:allWh)
        {
            System.out.println(text);
        }
        driver.quit();
    }
}

```

**Q: What is the difference between getWindowHandle() and getWindowHandles() ?**

getWindowHandle()	getWindowHandles()
Returns the address of current browsers	Returns the address of all the browsers.
Return type of getWindowHandle is String.	Return type of getWindowHandles() is a set of Strings.

**Q: WAS to get the titles of all the browsers opened by the naukri.com ?**

```
public class GetTitleOFChildWindows {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.naukri.com/");
        Set<String> allWh = driver.getWindowHandles();
        for(String text:allWh) {
            driver.switchTo().window(text); // here we are switching the
control to all the browsers opened.
            System.out.println(driver.getTitle()); // printing the title of the
browsers
        }
        driver.quit();
    }
}
```

**\*\*Q: WAS to close all the browsers present in naukri.com without using quit()?**

```
public class CloseAllChildWithoutQuit {
```

```

static{
    System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
}

public static void main(String[] args) throws InterruptedException {
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.naukri.com/");
    Set<String> allWh = driver.getWindowHandles();
    for(String text:allWh) {
        driver.switchTo().window(text);
        System.out.println(driver.getTitle());
        Thread.sleep(2000);
        driver.close();
    }
}
}

```

**\*\*Q: WAS to close all browsers except the parent browser in naukri.com?**

```

public class CloseOnlyChildBrowsers {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.naukri.com/");
        Set<String> allWh = driver.getWindowHandles();
        String parent = driver.getWindowHandle();
        for(String wh:allWh) {
            driver.switchTo().window(wh);
            if(parent.equals(wh)) {

            }
            else

```

```

        driver.close();
    }
}

```

**\*\*Q: WAS to close all the browsers except the specific browser?**  
**Or**  
**WAS to close only the specific browser?**

```

public class CloseSpecificBrowser {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        System.out.println("enter the title which you need to close");
        Scanner s = new Scanner(System.in);
        String expTitle = s.nextLine();
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.naukri.com/");
        Set<String> allWh = driver.getWindowHandles();
        for(String wh:allWh) {
            driver.switchTo().window(wh);
            String actualTitle = driver.getTitle();
            if(actualTitle.contains(expTitle)) //
if(actualTitle.equals(expTitle))
            {
                driver.close();
            }
        }
    }
}

```



## Summary of the PopUps

Popup	Solutions
Alert popup or javascript popup	switchTo().alert() statement
Hidden division or calendar popup	findElement()
File upload popup	browserbutton.sendKeys(absolute path)
File download popup	Robot class, for chrome no need to handle
Print popup	Robot class, for chrome we use findElement().
Notification popup	By using addArguments() of browser options class(ChromeOptions)
Authentication popup	By sending username and password along with URL in get().
Child window popup	By using getWindowHandles() and switchTo.window() statement.

## Handling tabs

In selenium Tab is also treated as a window. Here we are going to handle the tab in the same way as we handled child window or child browser popup.

**Q: WAS to count the number of tabs present in the browser after clicking actiTIME inc. link ?**

```
public class HandlingTabs {  
    static{  
        System.setProperty("webdriver.chrome.driver",  
            "./driver/chromedriver.exe");  
    }  
}
```

```

    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://demo.actitime.com/login.do");
        driver.findElement(By.linkText("actiTIME Inc.")).click();
        Set<String> allWh=driver.getWindowHandles();
        int count = allWh.size();
        System.out.println(count);
    }
}

```

**Q: how do you close the current tab?**

```
driver.close()
```

**Q: how do you close all the tabs?**

```
driver.quit()
```

**Q: How do you close all the tabs without using quit() after clicking the actiTIME Inc. link ?**

```

public class HandlingTabs {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://demo.actitime.com/login.do");
        driver.findElement(By.linkText("actiTIME Inc.")).click();
        Set<String> allWh=driver.getWindowHandles();
        int count = allWh.size();
        System.out.println(count);
        for(String allTabs:allWh) {
            driver.switchTo().window(allTabs);
            driver.close();
        }
    }
}

```

## Assignment

1. **WAS to print the title of all the tabs opened by the browser after clicking actiTIME Inc. ?**

```
public class actiTimeTitles {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://demo.actitime.com/login.do");
        driver.findElement(By.linkText("actiTIME Inc.")).click();
        Set<String> allWh=driver.getWindowHandles();
        for(String title:allWh) {
            driver.switchTo().window(title);
            System.out.println(driver.getTitle());
        }
    }
}
```

**\*Q: WAS to close the child tab 1st then the parent tab by using an iterator?**

```
public class HandlingTabsIterator {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.get("https://demo.actitime.com/login.do");
        driver.findElement(By.linkText("actiTIME Inc.")).click();
        Set<String> allWh=driver.getWindowHandles();
        Iterator<String> itr = allWh.iterator();
```

```

        String pwh = itr.next(); // allWh.iterator().next();
        String cwh = itr.next();
        Thread.sleep(2000);
        driver.switchTo().window(cwh);
        driver.close();
        driver.switchTo().window(pwh);
        driver.close();
    }
}

```

## Assignment

### 2. Automate the following scenarios

- a. open the browser.
- b. Go to actiTIME.com.
- c. Login to the application.
- d. Click on About your actiTIME present under Help dropdown.
- e. Click on Read Service Agreement link on the popup.
- f. Capture all the headings and print it on the console present on the new tab.
- g. Close the new tab.
- h. Close the parent tab.

```

public class actiTimeServiceAgreementHeadings {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.get("https://demo.actitime.com/login.do");
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        driver.findElement(By.xpath("//input[@type
='text']")).sendKeys("admin");

        driver.findElement(By.xpath("//input[@type='password']")).sendKeys("manager");
        driver.findElement(By.xpath("//div[text() ='Login '])).click();
    }
}

```

```

driver.findElement(By.xpath("//div[@class='popup_menu_icon']")[4])).click();
    driver.findElement(By.linkText("About your actiTIME")).click();
    Thread.sleep(2000);
    driver.findElement(By.xpath("//a[contains(., 'Read Service
Agreement')]")).click();
    Set<String> wh = driver.getWindowHandles();
    int count = wh.size();
    System.out.println(count);
    for(String agtab:wh) {
        driver.switchTo().window(agtab);
    }
    List<WebElement> headings =
driver.findElements(By.xpath("//h2"));
    for(WebElement allHeadings:headings)
    {
        System.out.println(allHeadings.getText());
    }
    driver.quit();
}
}

```

**3. WAS to cleartrip.com or yatra or makemytrip and search for the flights available on 30-Sep to Bangalore(BLR) to Goa(GOI), capture departure time and flight name and print it on the console after clicking on the search flight button.**

```

public class ClearTrip {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.cleartrip.com/");
    }
}

```

```

        driver.manage().timeouts().implicitlyWait(30,
TimeUnit.SECONDS);
        driver.findElement(By.xpath("//input[@placeholder='Any
worldwide city or airport']")[1])).sendKeys("BLR");

driver.findElement(By.xpath("//p[contains(., 'Bangalore')]")).click();
        driver.findElement(By.xpath("//input[@placeholder='Any
worldwide city or airport']")[2])).sendKeys("GOI");
        driver.findElement(By.xpath("//p[contains(., 'Goa')]")).click();
        driver.findElement(By.xpath("//div[@class='flex flex-middle
p-relative homeCalendar']")).click();
        driver.findElement(By.xpath("//div[@aria-label='Thu Sep 30
2021']")).click();
        driver.findElement(By.xpath("//button[.='Search
flights']")).click();
        List<WebElement> flights =
driver.findElements(By.xpath("//img[@class]"));
        List<WebElement> time =
driver.findElements(By.xpath("//img[@class]/../../../../../../../../div/div/div/div[2
]/div/div[1]/p"));
        int count = flights.size();
        int count1 = time.size();
        System.out.println(count+" "+ count1);
        for(int i=0;i<count;i++ ) {
            String allflights = flights.get(i).getAttribute("alt");
            String deptTime = time.get(i).getText();
            System.out.println("Flight_name: " + allflights
+"--->"+"Departure_time: "+deptTime);
        }
    }
}

```

## **Handling Mouse Actions:**

By using mouse actions we can perform following actions-

1. Handling mouse hover(dropdown menu→ move to element).
2. Right click- contextClick.
3. Drag and drop.
4. Double click.

### **Q: How do you handle mouse hover or drop down menu?**

Moving your mouse pointer to a particular location.

Drop down menu is an element on which if we move the mouse pointer it will display the list of options. To handle the drop down menu we use `moveToElement()` present in Actions class.

In selenium action is an interface and actions is class.

Actions class is present in the Interaction package and it is mainly used for mouse actions. Actions class as a parameterized constructor where it takes a `WebDriver` as an argument.

Whenever we call any methods of actions class we have to `perform()` at the end.

### **Q: Automate the following scenarios**

1. Open chrome browser.
2. Go to [Vtiger.com](https://vtiger.com)
3. Mouse hover to Resources tab and click on contact us in the drop down menu.
4. Get the Bengaluru phone number and print it on the console.
5. Close the browser.

```
public class MouseHover {  
    static{  
        System.setProperty("webdriver.chrome.driver",  
"./driver/chromedriver.exe");  
    }  
    public static void main(String[] args) {  
        WebDriver driver = new ChromeDriver();  
        driver.manage().window().maximize();  
    }  
}
```

```

        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        driver.get("https://www.vtiger.com/");
        Actions a = new Actions(driver);
        WebElement resources =
driver.findElement(By.partialLinkText("Resources"));
        a.moveToElement(resources).perform();
        driver.findElement(By.partialLinkText("Contact Us")).click();
        String phno =
driver.findElement(By.xpath("//p[contains(., 'Bengaluru')]/../p[2]")).getText();
        System.out.println(phno);
        driver.close();
    }
}

```

**Q: how do you perform a right click(contextClick) in selenium?**

Right clicking on the mouse is called a context link.

1. When we right click on any element we get a list of options which is called a context menu.
2. To right click on the element we use contextClick() of the actions class.
3. To select the required option in the context menu we press the shortcut keys such as w- new window, t- new tab etc.

**Q: WAS to open the actiTIME inc in the new window?**

```

public class RightClick {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) throws AWTException {
        WebDriver driver = new ChromeDriver();
        driver.get("https://demo.actitime.com/login.do");
        WebElement link = driver.findElement(By.partialLinkText("actiTIME
Inc."));
        Actions a = new Actions(driver);
        a.contextClick(link).perform();
        Robot r = new Robot();
    }
}

```



```

        r.keyPress(KeyEvent.VK_W);
    }
}

```

### **Q: How do you perform drag and drop in selenium?**

By using `dragAndDrop()` of actions class as shown in the below program

```

public class Drag_Drop {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();

        driver.get("http://www.dhtmlgoodies.com/submitted-scripts/i-google-like-drag-drop
/index.html");
        WebElement source = driver.findElement(By.xpath("//h1[.='Block
1']"));
        WebElement target = driver.findElement(By.xpath("//h1[.='Block
4']"));

        Thread.sleep(3000);
        Actions a = new Actions(driver);
        a.dragAndDrop(source, target).perform();
    }
}

```

### **Q: How do you perform a double click in selenium?**

By using `doubleClick()` of the actions class.

Example- `a.doubleClick(WebElement).perform();`

## Assignment

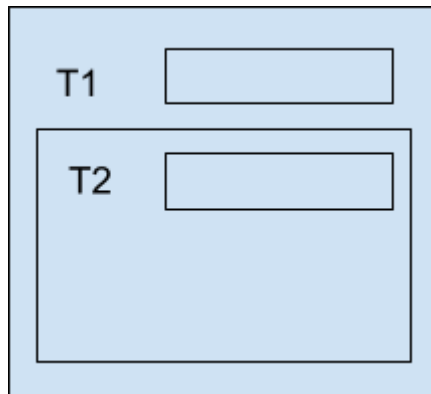
### Automate the following scenario

1. Go to Vtiger.com.
2. Click on customers present in the resources tab.
3. Double click on read full story button and check whether HackerEarth page is displayed or not.

```
public class MouseHoverVtiger {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.vtiger.com/");
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        Actions a = new Actions(driver);
        WebElement resource =
driver.findElement(By.partialLinkText("Resources"));
        a.moveToElement(resource).perform();
        driver.findElement(By.partialLinkText("Customers")).click();
        driver.manage().window().maximize();
        WebElement frs = driver.findElement(By.linkText("READ FULL
STORY"));
        a.doubleClick(frs).perform();
        String t1= driver.getTitle();
        String t2= "HackerEarth Case Study | Vtiger CRM";
        if(t1.equals(t2))
            System.out.println("you are on the right page");
        else
            System.out.println("you are on the wrong page");
        driver.quit();
    }
}
```

**Note:** all the methods of the actions class are examples of method overloading.

### Handlings Frames



- A web page inside another web page is called an embedded web page (frames).
- Developers create embedded web pages using the iframe tag.

HTML code to create page1 is

T1 :<input type="text" id="t1"/><br>  
<iframe src="page2.html" id="f1"/>

HTML code to create page2 is

T2: <input type="text" id="t2"/>

**Q: WAS to enter jsp in T2 textbox and qsp in T1 textbox ?**

```
public class HandlingFrames {  
    static{  
        System.setProperty("webdriver.chrome.driver",  
"./driver/chromedriver.exe");  
    }  
    public static void main(String[] args) {  
        WebDriver driver = new ChromeDriver();  
        driver.get("file:///C:/Users/SATEESH/Desktop/page1.html");  
    }  
}
```

```

        driver.switchTo().frame(0);
        driver.findElement(By.id("t2")).sendKeys("jsp");
        driver.switchTo().parentFrame();
        driver.findElement(By.id("t1")).sendKeys("qsp");
    }
}

```

**Q: WAS to type ABC in T2 textbox and DE in T1 textbox character by character alternatively ?**

```

public class HandlingFrames {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/page1.html");
        driver.switchTo().frame(0);
        driver.findElement(By.id("t2")).sendKeys("a");
        driver.switchTo().parentFrame();
        driver.findElement(By.id("t1")).sendKeys("d");
        driver.switchTo().frame("f1");
        driver.findElement(By.id("t2")).sendKeys("b");
        driver.switchTo().parentFrame();
        driver.findElement(By.id("t1")).sendKeys("e");
        WebElement frame = driver.findElement(By.xpath("//iframe"));
        driver.switchTo().frame(frame);
        driver.findElement(By.id("t2")).sendKeys("c");
    }
}

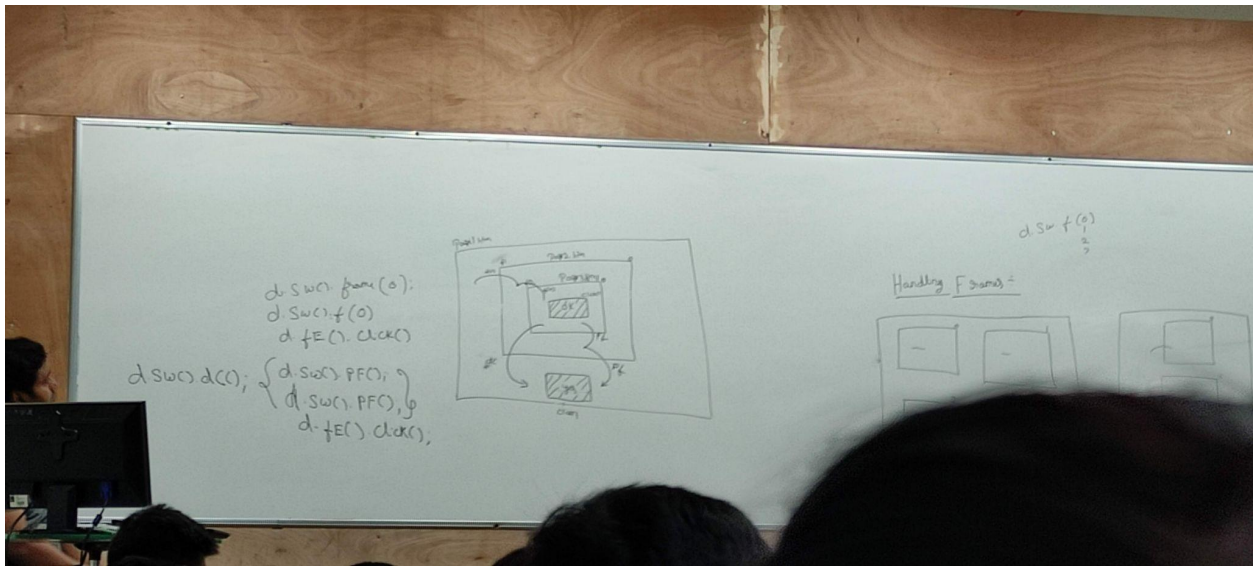
```

In the above example frame() is overloaded, it takes only one argument of any of the following types

1. Int (index of the frame and starts from 0)
2. String (id or attribute of the frame)
3. WebElement (address of the frame)

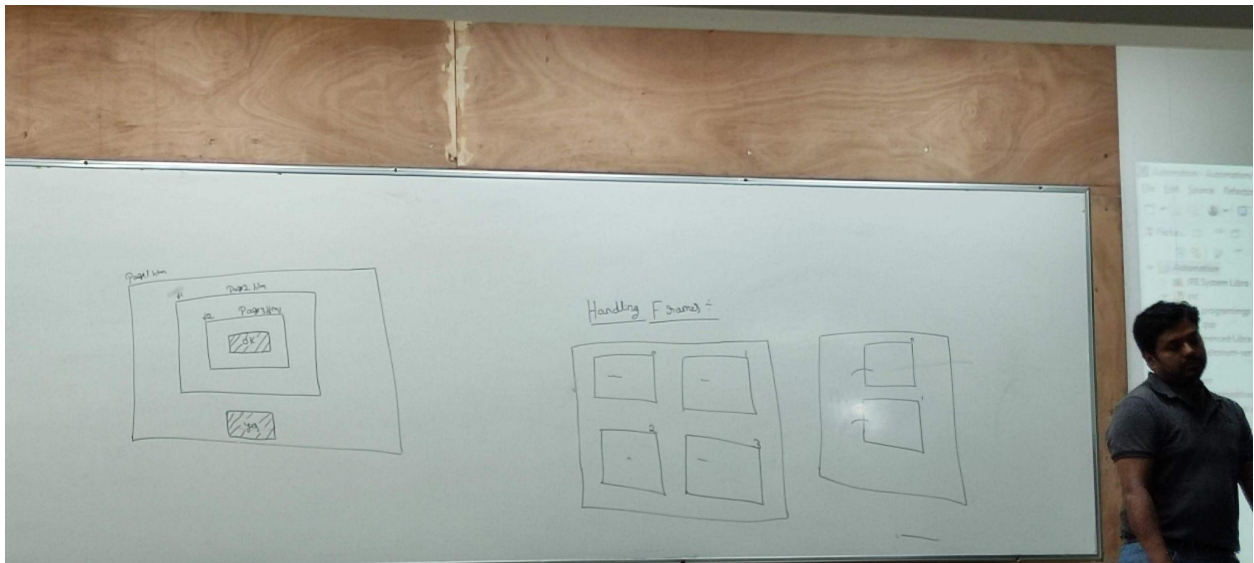
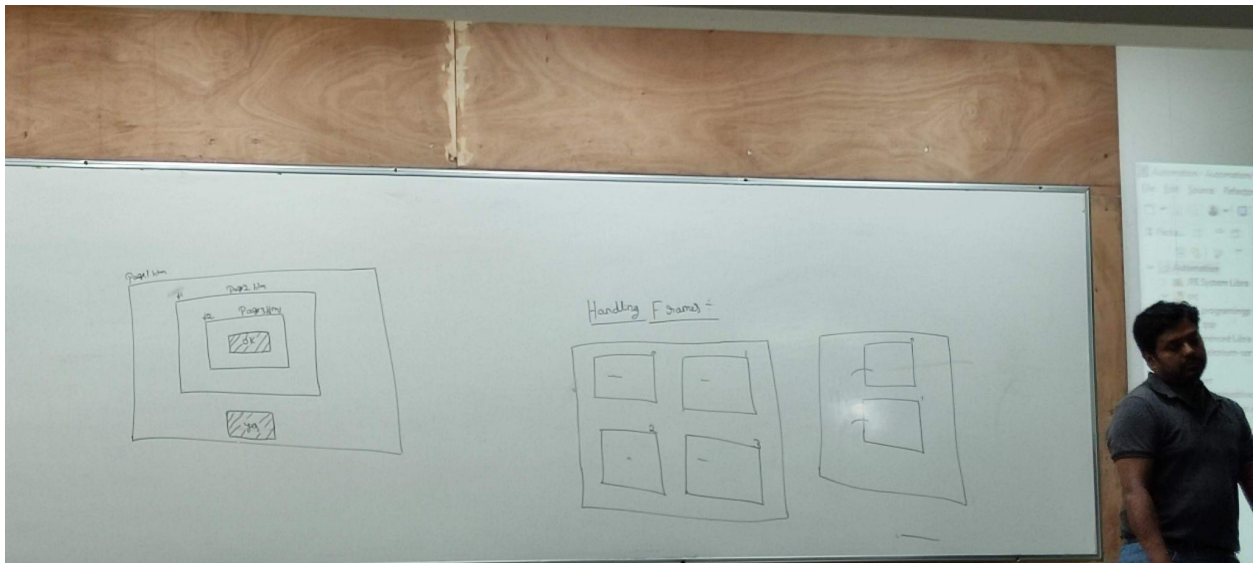
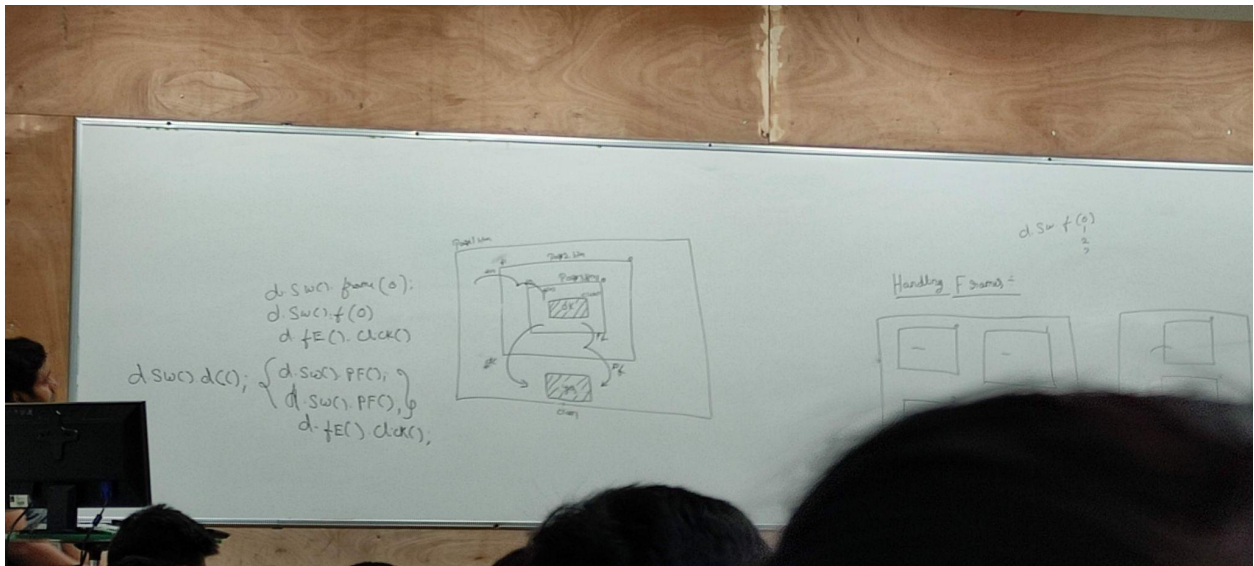
$dsw(f, a)$   
 $dsw(f, a)$   
 $dfe(1, clock)$   
 $dsw(dcc); \left\{ \begin{array}{l} dsw(pfc); \\ dsw(pfc); \\ dfe(1, clock); \end{array} \right\}$

Handling Functions:



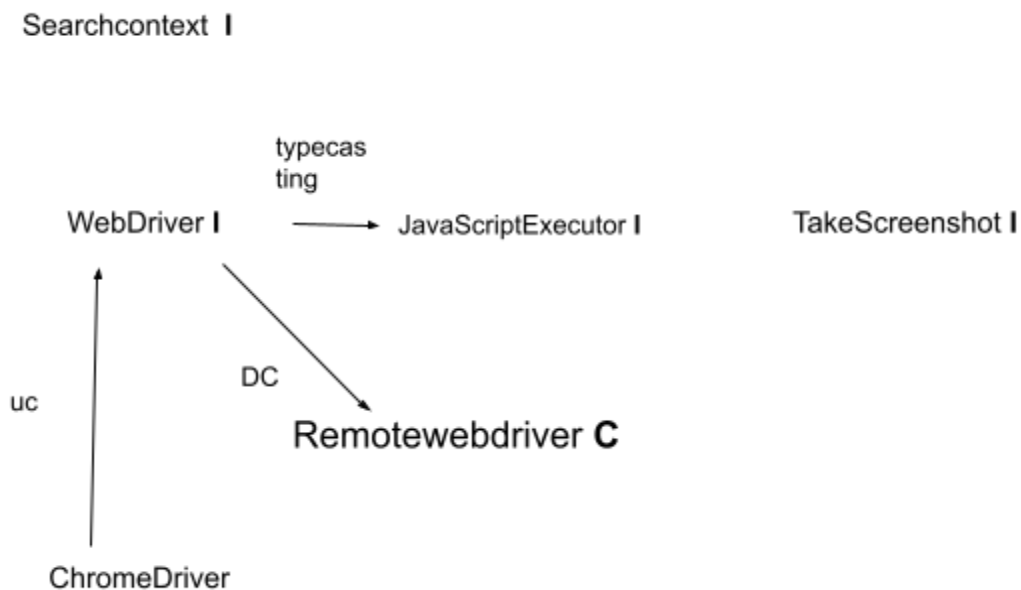
$dsw(f, a)$   
 $dsw(f, a)$   
 $dfe(1, clock)$   
 $dsw(dcc); \left\{ \begin{array}{l} dsw(pfc); \\ dsw(pfc); \\ dfe(1, clock); \end{array} \right\}$

Handling Functions:



## Handling Disable elements and Scroll bar

- In selenium we don't have a method to handle disabled elements or scrollbar (by using java language).
- In order to handle it we use `executeScript()`. The `executeScript()` is declared in the `JavaScriptExecutor` interface. Since we already upcasted browser specific classes to the `WebDriver` interface this method will be hidden.
- In order to access this method we should downcast to `RemoteWebDriver` or typecast it to `JavaScriptExecutor`.



HTML code for static webpage

```
UN:<input id="d1" type="text"/><br>
PW:<input id="d2" tpye="text" disabled/><br>
<input type="button" value="login"/>
```

### **Q: WAS to enter the manager in the disabled password textbox?**

```
public class HandlingDisabledElements {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/disabled.html");
        driver.findElement(By.id("d1")).sendKeys("admin");
        RemoteWebDriver r= (RemoteWebDriver)driver;
        r.executeScript("document.getElementById('d2').value ='manager'");
    }
}
```

### **Note**

In order to validate the javascript program inspect the element and click on the console tab in developer toolbar and type this script  
document.getElementById('d2').value ='manager'

For clicking a button: document.getElementById(d3).click();

For deleting the text: document.getElementById(d2).value=' ' ;

### **Scrolling**

### **Q: WAS to scroll 4000 pixels vertically in BBC.com.**

```
public class HandlingScrollBar {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.bbc.com/");
    }
}
```

```

        JavascriptExecutor j = (JavascriptExecutor) driver;
        j.executeScript("window.scrollTo(0,4000)");
    }
}

```

**Q: WAS to scroll to the particular element in the BBC.com .**

```

public class HandlingScrollBar {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.bbc.com/");
        int y=driver.findElement(By.xpath("//span[.='Future
Planet']")).getLocation().getY();
        JavascriptExecutor j = (JavascriptExecutor) driver;
        j.executeScript("window.scrollTo(0,\"+y+\")"); // we need to
concatenate y, else it will take it as a string
    }
}

```

**Q: WAS to scroll to the bottom of the webpage in BBC.com ?**

```

public class ScrollToBottom {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.bbc.com/");
        JavascriptExecutor j = (JavascriptExecutor) driver;
        // scroll to bottom of the page
        j.executeScript("window.scrollTo(0,document.body.scrollHeight)");
        Thread.sleep(2000);
    }
}

```



```

        // scroll to top of the page
        j.executeScript("window.scrollTo(0,0)");
    }
}

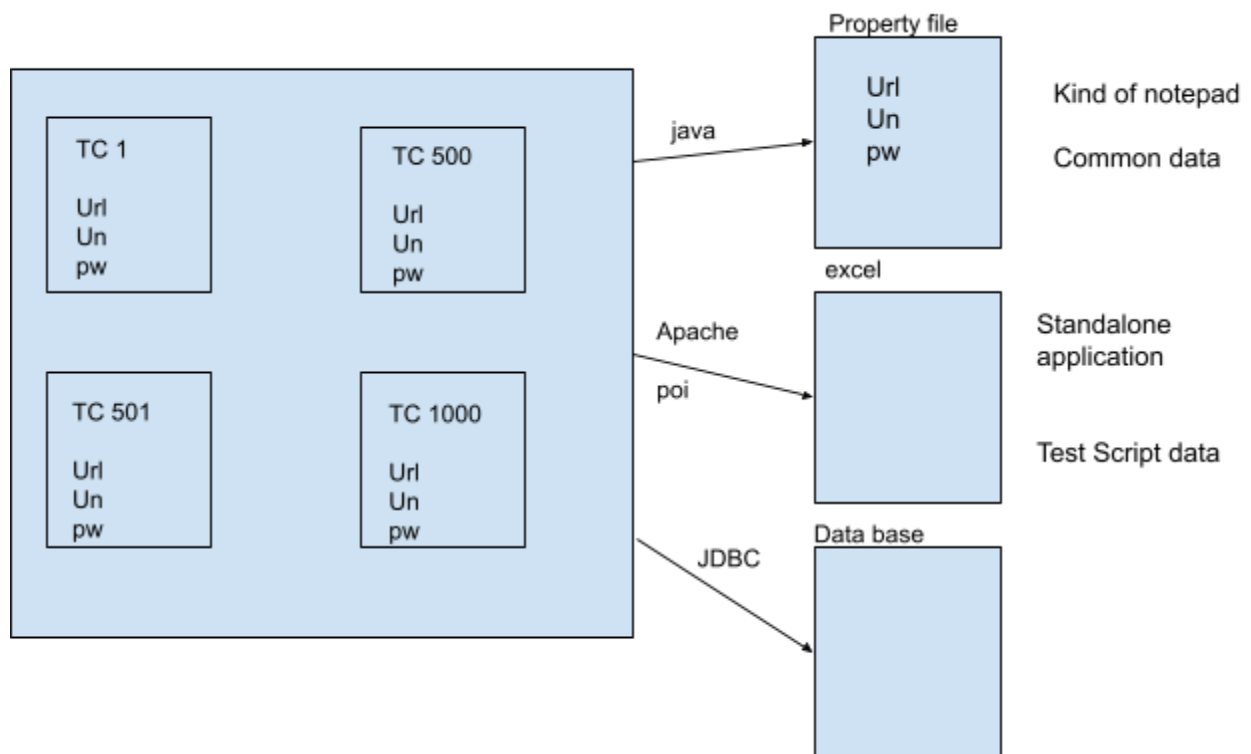
```

### Note

scrollTo - it will scroll or it will consider from the top of the webpage always.  
 scrollBy- it will scroll from wherever the scroll bar or control is present.

### Data driven testing

Testing the application with multiple data which is kept in external resource files like Excel, Property file, Database etc. is called data driven testing.



```

public class DemoMaps {

    public static void main(String[] args) {
        ArrayList<String> lst = new ArrayList<String>();
        lst.add("https://demo.actitime.com");
        lst.add("admin");
        lst.add("admin@actitime.com"); // if email is added in future
        lst.add("manager");
        System.out.println(lst.get(2)); // it will print emial instead of
password- manager

        HashMap<String, String> hs = new HashMap<String, String>();
        hs.put("url", "https://demo.actitime.com");
        hs.put("username","admin");
        hs.put("email", "admin@actitime.com");
        hs.put("password","manager");
        System.out.println(hs.get("password")); // whether mail is added or
not in future, it will print same output
    }
}

```

LIST

URL	0
Username	1
Password	2

LIST

URI	0
Username	1
<a href="mailto:admin@actitime.com">admin@actitime.com</a>	2
Password	3

## Note

- If we use index we will not retrieve the same data if any data is added or deleted tomorrow.
- we get the same output if new data is added or deleted in future, if we use a Key-Value pair of matches.
- Same concept is used to store data in property files.
- As per the rule of automation test data should not be Hardcoded within the test script, because modification and maintenance of test data is difficult. So we use external resource files like property files, excel, databases etc.

## Handling property files

- Property file is used to store the common data. In order to create the property file enter the below data in the notepad and save it as any filename with .property extension.

url <https://demo.actitime.com>  
username admin  
password manager  
email [admin@actitime.com](mailto:admin@actitime.com)

- In property file data will be stored in the form of Key-Value pair, KEY and VALUE should be separated by single space.
- By default all the data available in the property file is a String.

```
public class HandlingPropertyFiles {  
  
    public static void main(String[] args) throws IOException {  
        // get the java respective object of physical file  
        FileInputStream fis = new  
FileInputStream("./data/commondata.property");  
        // create an object of properties class  
        Properties p = new Properties();  
        //load the file so that getProperty method will come to know where  
the file is
```

```

        p.load(fis);
        // read or get the data from the property file by passing the key
        String url = p.getProperty("url");
        String email = p.getProperty("email");
        String pw = p.getProperty("password");
        String un = p.getProperty("username");
        System.out.println(url);
        System.out.println(pw);
    }
}

```

**\*Q: WAS to login to actiTIME application by taking the test data from the property file.**

```

public class actiTimePropertyFile {
    static{
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) throws IOException {
        WebDriver driver = new ChromeDriver();
        driver.get("https://demo.actitime.com/login.do");
        FileInputStream fis = new
FileInputStream("./data/commondata.property");
        Properties p = new Properties();
        p.load(fis);
        String un = p.getProperty("username");
        String pw = p.getProperty("password");
        driver.findElement(By.xpath("//input[@type='text']")).sendKeys(un);
        driver.findElement(By.xpath("//input[@type='password']")).sendKeys(pw);
        driver.findElement(By.xpath("//div[text()='Login']")).click();
    }
}

```

## **Advantages of Property file-**

1. It is very fast in execution.
2. It is very light weight compared to any other external resource files.

## **Handling data driven from Excel file**

Whenever we want to read the data from Excel we need to use Apache POI plugin or third party tool, so that we can read or write the data from all microsoft documents like .xls, .xlsx, .docx, .ppt, .outlook etc.

-Apache poi is a free and open source library tool similar to selenium.

## **Steps to install Apache POI**

1. Go to the URL : <https://poi.apache.org/download.html>
2. Under Binary distribution - Click on poi-bin-5.0.0.... Zip file
3. Click on the first link  
(<https://mirrors.estointernet.in/apache/poi/release/bin/poi-bin-5.0.0-20210120.zip>)
4. Zip file will download.
5. Extract the zip file.
6. Copy all the jars present in every folder and paste it inside the jar folder of the eclipse.
7. Add all the jars to the build path.

**\*Q: WAS to read the data from the excel ?**

```
public class HandlingExcel {  
  
    public static void main(String[] args) throws EncryptedDocumentException,  
IOException {  
        // get the java representative object of physical file  
        FileInputStream fis = new FileInputStream("./data/testscript.xlsx");  
        // load the file or create a workbook  
        Workbook wb = WorkbookFactory.create(fis);  
        //get the control of the sheet, get the control of row, then cell, the get  
or read the data  
        String data =  
wb.getSheet("CreateCustomer").getRow(1).getCell(2).getStringCellValue();  
    }  
}
```

```

        System.out.println(data);
    }
}

```

**\*Q: WAS to write the data back into the Excel ?**

```

public class WritingDataInExcel {

    public static void main(String[] args) throws EncryptedDocumentException,
IOException {
        FileInputStream fis = new FileInputStream("./data/testscript.xlsx");
        Workbook wb = WorkbookFactory.create(fis);
        // get the control of the sheet, the row, then cell, the type the value

        wb.getSheet("CreateCustomer").getRow(1).getCell(4).setCellValue("fail");
        // convert the java representative object into physical file format
        FileOutputStream fos = new
        FileOutputStream("./data/testscript.xlsx");
        // save the workbook or file (actual writing happens here)
        wb.write(fos);
        // close the workbook or file
        wb.close();
    }
}

```

### **Note**

- Whenever we are reading the data from the excel the extension should be .xlsx
- All the apache file classes should be imported from org.apache.poi.ss.usermodel package.
- While writing the data back to excel, a specific excel file should be closed, i.e, you should close the file before doing any modification.

**Q: WAS to read multiple data from the excel ?**

```

public class HandlingMultipleExcelData {

```

```

        public static void main(String[] args) throws EncryptedDocumentException,
IOException {
            FileInputStream fis = new FileInputStream("./data/testscript.xlsx");
            Workbook wb = WorkbookFactory.create(fis);
            // get last row number where the data is present
            int rowcount = wb.getSheet("InvalidLogin").getLastRowNum();
            for(int i=1;i<= rowcount;i++) {
                String un =
wb.getSheet("InvalidLogin").getRow(i).getCell(0).getStringCellValue();
                String pw =
wb.getSheet("InvalidLogin").getRow(i).getCell(1).getStringCellValue();
                System.out.println(un+ "---->" + pw);
            }
        }
    }
}

```

## Assignment

### 1. Do the above program using nested for(i,j)

```

        public static void main(String[] args) throws EncryptedDocumentException,
IOException {
            FileInputStream fis = new FileInputStream("./data/testscript.xlsx");
            Workbook wb = WorkbookFactory.create(fis);
            // get last row number where the data is present
            int rowcount = wb.getSheet("InvalidLogin").getLastRowNum();
            short cellcount =
wb.getSheet("InvalidLogin").getRow(0).getLastCellNum();
            for(int i=1;i<= rowcount;i++) {
                for(int j=0;j<cellcount;j++) {
                    String un =
wb.getSheet("InvalidLogin").getRow(i).getCell(j).getStringCellValue();
                    System.out.println(un);
                }
            }
        }
    }
}

```

**Q: Another way to read the data from the excel**

```
public class HandlingExcel {  
    public static void main(String[] args) throws EncryptedDocumentException,  
        IOExcept {  
        FileInputStream fis=new FileInputStream("./data/testscript.xlsx");  
        Workbook wb = WorkbookFactory.create(fis);  
        //get the control of the sheet  
        Sheet sheet = wb.getSheet("CreateCustomer");  
        //get the control of the row  
        Row row = sheet.getRow (1);  
        //get the control of the cell Cell cell = row.getCell(2);  
        //get the data from the cell  
        String data = cell.getStringCellValue();  
        //print it on the console  
        System.out.println(data);  
    }  
}
```

**Advantages of Data driven testing:**

1. Maintenance of the test data in excel or property file is easier.
2. Modification of the test data in excel or external resource file is easier.
3. Reusability of test script data and common data.
4. Test data can be created before the test execution.
5. We can test the application with a huge volume of data.

**Generic library**

- Generic library is one of the components in the automation framework. It contains common classes and methods which can be reusable for all the test scripts and in any project as well.
- Generic library classes contain reusable methods which are created by framework developers.
- All the generic classes should be created in separate package and the name of the package should be *com.projectname.generic*



- Whenever we are creating any generic classes and methods it should follow the coding standards as shown in the below programs.

```
package com.sateesh.generic;
```

```
import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.util.Properties;
```

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

```
/**
```

```
 * this is a generic class for data driven testing
```

```
 * @author SATEESH
```

```
 *
```

```
 */
```

```
public class FileLib {
```

```
    /**
```

```
     * This is the generic method to read the data form the property file
```

```
     * @param args
```

```
     * @throws IOException
```

```
    */
```

```
    public String getPropertyData(String key) throws IOException {
```

```
        FileInputStream fis = new
```

```
        FileInputStream("./data/commonddata.property");
```

```
        Properties p = new Properties();
```

```
        p.load(fis);
```

```
        String data = p.getProperty(key);
```

```
        return data;
```

```
    }
```

```
    /**
```

```
     * this is the generic method to read the data from Excel file
```

```
     * @throws IOException
```

```

    * @throws EncryptedDocumentException
    *
    */
    public String getExcelData(String sheetname, int row, int cell) throws
EncryptedDocumentException, IOException {
        FileInputStream fis = new FileInputStream("./data/testscript.xlsx");
        Workbook wb = WorkbookFactory.create(fis);
        String data =
wb.getSheet(sheetname).getRow(row).getCell(cell).getStringCellValue();
        return data;
    }
    /**
    * this is the generic method to write the data into the Excel file
    * @param sheetname
    * @param row
    * @param cell
    * @param value
    * @throws EncryptedDocumentException
    * @throws IOException
    */
    public void setExcelData(String sheetname,int row, int cell, String value)
throws EncryptedDocumentException, IOException {
        FileInputStream fis = new FileInputStream("./data/testscript.xlsx");
        Workbook wb = WorkbookFactory.create(fis);

wb.getSheet(sheetname).getRow(row).getCell(cell).setCellValue(value);
        FileOutputStream fos = new
FileOutputStream("./data/testscript.xlsx");
        wb.write(fos);
        wb.close();
    }
}

```

In order to **call the generic method** refer the below program:

```

public class DemoGeneric {

    public static void main(String[] args) throws IOException {
        FileLib f = new FileLib();
        // to read the data from property file
        System.out.println(f.getPropertyData("url"));
        // to write the data to excel file
        f.setExcelData("CreateCustomer", 1, 4, "skipped");
        // to read the data from the excel file
        System.out.println(f.getExcelData("CreateCustomer", 1, 4));
    }
}

```

### **Webdriver common library**

```

package com.sateesh.generic;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.support.ui.WebDriverWait;

public class WebDriverCommonLib {
    /**
     * generic method for implicit wait
     * @param driver
     * @param time
     */
    public void waitForElementToLoad(WebDriver driver, int time){
        driver.manage().timeouts().implicitlyWait(time, TimeUnit.SECONDS);
    }
}

```

```

/**
 * generic method for explicit wait until element is visible
 * @param driver
 * @param time_value
 * @param element
 */
public void waitForElementInGui(WebDriver driver, int time_value,
WebElement element) {
    WebDriverWait wait = new WebDriverWait(driver, time_value);
    wait.until(ExpectedConditions.visibilityOf(element));
}
/**
 * generic method for custom wait till element is displayed
 * @param element
 */
public void customWaitForElement(WebElement element) {
    int i =0;
    while(i<1000) {
        try {
            element.isDisplayed();
            break;
        }
        catch(Exception e) {
            i++;
        }
    }
}
/**
 * select the value from the textbox based on the text
 * @param element
 * @param text
 */
public void select(WebElement element, String text) {
    Select s = new Select(element);
    s.selectByVisibleText(text);
}
/**

```

```

    * select value from the textbox based on the index
    * @param element
    * @param i
    */
    public void select(WebElement element, int i) {
        Select s = new Select(element);
        s.selectByIndex(i);
    }
}

```

## **Unit Testing Framework Tool**

1. J unit (java)
2. N unit (.net)
3. PyDev (python)
4. RSPC (Ruby)
5. testNG

## **TestNG (test Next Generation)**

testNG is a unit testing framework tool which is mainly used for Batch execution. Basically testNG is used by the developers to perform unit testing and it is also used by automation engineers to perform black box testing

Or

testNG is a plugin for eclipse which is inspired by Junit and Nunit with some additional features.

## **Advantages or additional features of testNG**

1. Batch execution
2. Parallel execution
3. Group execution
4. Generate reports automatically (HTML report)
5. Additional annotations
6. Execute only failed test scripts
7. Listener features

### **Installation of testNG (use 2019 and above eclipse version)**

1. In eclipse go to Help and select Eclipse Marketplace.
2. Type testNG in the search textbox and click on the Search button.
3. Click on the install button present under testNG for eclipse.
4. Click on confirm.
5. Click on the radio button "I agree to license agreement".
6. Click on next and click on Install anyway on the Popup.
7. Click on finish and restart now.

### **Another way of installing testNG**

1. Go to Help and select install new software.
2. Type this URL <https://testng.org/testng-eclipse-update-site> in the work with text box and click on enter.
3. Select the testNG checkbox and click on next.
4. Remaining steps are the same as above.

### **Note**

After the TestNG installation to the eclipse, add the TestNG library (jar) to the java project as well.

- right click on the java project, go to Build path and select Add Libraries.
- select testNG and click on Next and Finish.
- it will display the TestNG folder in the java project.

While creating TestNG classes we should not use following things:

1. Don't use the default package.
2. Don't use the main method.
3. Don't use System.out.println statement.

```
package com.actitime.testscript;
```

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

```
public class Demo {  
    @Test  
    public void testDemo() {  
        Reporter.log("welcome to testng",true);  
    }  
}
```

}

- When we execute the above code it automatically generates the report in HTML format.

In order to see it do the following steps

1. Right click on the project and refresh it.(f5)
2. Expand the test output folder and right click on emailabe-report.html.
3. Go to open with and select the web browser.

In order to get report in excel format

1. Right click on the report and select export to microsoft excel.
2. Click on import and click on ok button.

*System.out.println("hi"); -----> print only on the console*

*Reporter.log("welcome to testng"); -----> print it on the console and report also*

*Reporter.log("bye",false); -----> print only on the report*

*Reporter.log("hello"); -----> print only on the report*

**Q.If a class contains multiple test methods in which order they are executed?**

Alphabetical order.

**Q: How to execute the test method in required order?**

Using priority

syntax - @Test(priority =1)

Default value for priority is 0.

If priority is duplicate then those methods will be executed in alphabetical order.

We can specify -ve value in priority and it will execute them in ascending order.

Variables and decimals are not allowed.

**Q: How do you run a method multiple times?**

By using invocationCount

Syntax- @Test(invocationCount = 7)

Default value of invocationCount is 1.

If we specify 0 or -ve value it will not execute that test method.  
Variables and decimals(fractions) are not allowed.

### **Q: How do you make a test depend on another test ?**

Using dependsOnMethods

Syntax- `@Test(dependsOnMethods= "another test name")`

For multiple

`@Test(dependsOnMethods={ "another test name", "test name 2" })`

- If both dependency and priority are specified it will consider the dependency.

### **Q: What if two methods are dependent on each other?**

We get `TestNGException` (error- cyclic dependencies)

### **Example:**

```
public class Demo1 {  
    @Test(priority=1,invocationCount=1)  
    public void createCustomer() {  
        Reporter.log("createCustomer",true);  
    }  
  
    @Test(priority =3)  
    public void modifyCustomer() {  
        Reporter.log("modifyCustomer",true);  
    }  
  
    @Test(priority =2, dependsOnMethods="modifyCustomer")  
    public void deleteCustomer() {  
        Reporter.log("deleteCustomer",true);  
    }  
}
```

### Output:

createCustomer  
modifyCustomer  
deleteCustomer



## **Q: How do you disable the test case or test method ?**

By using enabled = false

Syntax- @Test(enabled = false)

## **Batch Execution**

Executing multiple test classes or test scripts is called batch execution.

Or

Executing all of them together via testng.xml file is called batch execution.

## **TestNG suit**

- Testng suit is an xml file which contains a list of testng classes (test classes) which are to be executed.
- Testng classes are the test classes which means a class contains @Test annotation or test method.
- Test methods means a method which has @Test annotation is called test method.
- To create the suit file follow the below steps:
  1. Write click on java project.
  2. Go to testng and select convert to testng.
  3. Click on finish. It will create a suit file called testng.xml .

To execute it:

1. Right click on testng.xml file (suit file).
2. Go to run as and select testng suit.

Or

1. Open the suit file and click on the run button.

## **Content of testng.xml file**

```
<suite name="Suite">
  <test thread-count="5" name="Test">
    <classes>
      <class name="qsp.DemoA"/>
      <class name="com.actitime.testscript.ProjectModule"/>
      <class name="com.actitime.testscript.Demo"/>
      <class name="com.actitime.testscript.CustomerModule"/>
      <class name="com.actitime.testscript.TaskModule"/>
    </classes>
  </test>
</suite>
```

```
</classes>
</test>
</suite>
```

**Q: How do you execute only the failed test cases or test methods?**

By using testng-failed.xml file present under test output folder.

**Q: How do you fail the test case intentionally?**

By using assert.fail

**Example for test class**

```
public class TaskModule {
    @Test(priority =1,invocationCount=3)
    public void createTask() {
        Reporter.log("createTask",true);
    }

    @Test(dependsOnMethods ="createTask")
    public void modifyTask() {
        Reporter.log("modifyTask",true);
    }

    @Test(dependsOnMethods= {"createTask","modifyTask"})
    public void deleteTask() {
        Reporter.log("deleteTask",true);
    }
}
```

Output

```
createTask
createTask
createTask
modifyTask
deleteTask
```

## Assertion

Assertion is one of the features available in testng which is used to verify the expected result of the test script.

```
public class Assertion {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    @Test
    public void testVerifyTitle()
    {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com/");
        String expectedTitle = "Sooogle";
        String actualTitle = driver.getTitle();
        if(actualTitle.equals(expectedTitle)) {
            Reporter.log("title is matching so pass", true);
        }
        else {
            Reporter.log("title is not matching so fail", true);
        }
        driver.close();
    }
}
```

## Note

As per the rule of automation every expectation should be verified with assert statement instead of java if else, because if else doesn't have the capacity to fail the test script.

There are 2 types of assert statement available in testng

1. asser(HardAssert)
2. SoftAssert

**Q: What are the important methods present in assertion or assert class?**

1. fail()
2. assertEquals()

3. assertEquals()
4. assertSame()
5. assertNotSame()
6. assertTrue()
7. assertFalse()
8. assertNull()
9. assertNotNull()

All the above methods are static methods of assert class.

**Q: How do you compare actual and expected values without using equals()?**

Assert.assertEquals(actualTitle, expectedTitle);

```
public class Assertion {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
@Test
public void testVerifyTitle()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.google.com/");
    String expectedTitle = "Soogle";
    String actualTitle = driver.getTitle();
    Assert.assertEquals(actualTitle, expectedTitle);
    driver.close();
}
}
```

**Note**

\* If the comparison fails then the statements which are present after the Assert statement of the current test method will not be executed.

\*In the above example it will not close the browser if the comparison fails.

\*In order to continue the execution even after failure of the comparison, we can use SoftAssert, but here all the methods are non static (both have same number of methods except *assertAll()* )

```

public class Assertion {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
@Test
public void testVerifyTitle()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.google.com/");
    String expectedTitle = "Sooogle";
    String actualTitle = driver.getTitle();
    SoftAssert s = new SoftAssert();
    s.assertEquals(actualTitle, expectedTitle);
    driver.close();
    s.assertAll();
}
}

```

### Note

- \* To update the status of the information into the result window we should use assertAll() at the last.
- \* Any statements after assertAll() will not be executed if the comparison fails.

### \*Q: Difference b/w Assert and SoftAssert .

Assert (verify)	SoftAssert
1.All the methods are static	1.All the methods are non static.
2.If the comparison fails, the remaining statement will not be executed in the current method.	2.Executes remaining statements even if the comparison fails.
3.We don't call assertAll()	3. We should call assertAll() at the end.

## **Note**

All the methods present in the assert class are examples for method overloading.

## **Annotation**

Annotation is the metadata which is used to provide special instruction to the java compiler during run time.

## **Important annotations of TestNG**

@BeforeSuite  
@BeforeTest  
@BeforeClass  
@BeforeMethod  
@Test  
@AfterSuite  
@AfterTest  
@AfterClass  
@AfterMethod

## **Optional annotations**

@DataProvider  
@Listeners  
@Parameters  
@AfterGroups  
@BeforeGroups  
@Factory  
@ignore -----> it will ignore the test script --skips  
@Notifications  
@ObjectFactory  
@TestInstance

## **Annotation usage in real time**

Program:

```
public class CustomerModule {  
    @BeforeMethod  
    public void login() {  
        Reporter.log("login",true);  
    }  
}
```

```

}
@AfterMethod
public void logout() {
    Reporter.log("logout",true);
}
@Test
public void createCustomer() {
    Reporter.log("createCustomer",true);
}
@Test
public void deleteCustomer() {
    Reporter.log("deleteCustomer",true);
}
@Test
public void modifyCustomer() {
    Reporter.log("modifyCustomer",true);
}
}

```

### Output

```

login
createCustomer
logout
login
deleteCustomer
logout
login
modifyCustomer
logout

```

---

```

public class CustomerModule {
    @BeforeClass
    public void openBrowser() {
        Reporter.log("openBrowser",true);
    }
    @AfterClass
    public void closeBrowser() {

```

```

        Reporter.log("closeBrowser",true);
    }
    @BeforeMethod
    public void login() {
        Reporter.log("login",true);
    }
    @AfterMethod
    public void logout() {
        Reporter.log("logout",true);
    }
    @Test(priority=1,invocationCount=2)
    public void editCustomer() {
        Reporter.log("editeCustomer",true);
    }
    @Test
    public void registerCustomer() {
        Reporter.log("registerCustomer",true);
    }
    @Test
    public void deleteCustomer() {
        Reporter.log("deleteCustomer",true);
    }
}

```

### Output

```

openBrowser
login
deleteCustomer
logout
login
registerCustomer
logout
login
editeCustomer
logout
login
editeCustomer
logout

```



closeBrowser

### **@BeforeMethod**

BeforeMethod annotation will be executed before executing every test method(@Test) present in the class.

In real time the before method will be used to develop common pre conditions which is required for all the test scripts like login programs.

### **@AfterMethod**

AfterMethod annotation will be executed after executing every @Test method in a class.

In real time it is used to develop common post conditions like logout programs.

### **@BeforeClass**

BeforeClass annotation will be executed before executing every test class.

In real time, a before class method will be used to open the browser.

### **@AfterClass**

AfterClass annotation will be executed after every @Test class.

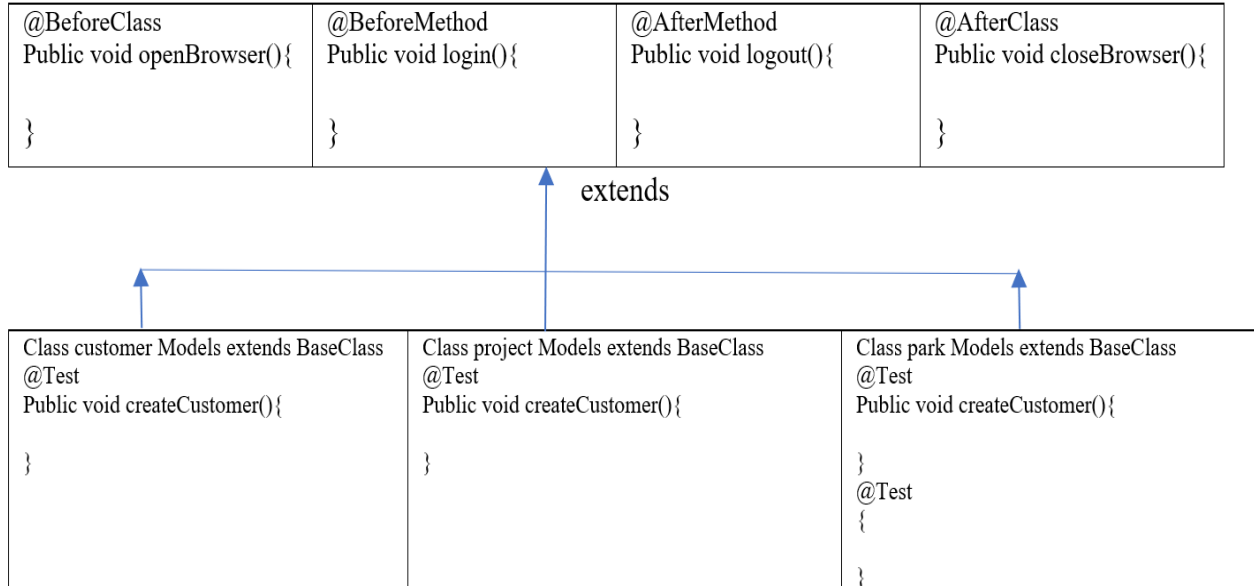
In real time it is used to close the browser.

### **@Test**

Test annotation indicates the test method.

In real time one test method represents one test script or test case.

## BASE CLASS



\*BaseClass is the supermost class in the framework which is created under com.sateesh.generic package. It contains configuration methods like open browser, login ,logout and close browser etc. all the test classes should be created under the com.actitime.testscript package.

\*Every test class should extend the base class.

Example : public class CustomerModule extends BaseClass

Generic code:

```
public class BaseClass {  
    @BeforeClass  
    public void openBrowser() {  
        Reporter.log("openBrowser",true);  
    }  
    @AfterClass  
    public void closeBrowser() {  
        Reporter.log("closeBrowser",true);  
    }  
    @BeforeMethod  
    public void login() {  
        Reporter.log("login",true);  
    }  
}
```

```

    }
    @AfterMethod
    public void logot() {
        Reporter.log("logout",true);
    }
}

```

### **Q: WAS to take a screenshot of google.com ?**

```

public class ScreenShot {
    static {
        System.setProperty("webdriver.chrome.driver",
            "./driver/chromedriver.exe");
    }
    @Test
    public void testScreenShot() throws IOException {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com/");
        TakesScreenshot t= (TakesScreenshot) driver;
        File src = t.getScreenshotAs(OutputType.FILE);
        File dest = new File("./Screenshot/ss.png");
        FileUtils.copyFile(src,dest);
        driver.close();
    }
}

```

### **Note**

Inorder to use FileUtils.copyFile statement we need to download commons io jar from apache commons community

([https://commons.apache.org/proper/commons-io/download\\_io.cgi](https://commons.apache.org/proper/commons-io/download_io.cgi))

Click on the above link and under Binaries click on the second link

(<https://downloads.apache.org/commons/io/binaries/commons-io-2.11.0-bin.zip>)

Extract the zip file and copy the 1st jar file and paste it in the eclipse jar folder.

Add that jar file to the build path.

## Assignment

Use generic code in base class to enter UN and Pwd in actiTime.com

```
public class GenericBaseClass {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static WebDriver driver;
    @BeforeClass
    public void openBrowser() {
        Reporter.log("openBrowser",true);
        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    }
    @AfterClass
    public void closeBrowser() {
        Reporter.log("closeBrowser",true);
        driver.close();
    }
    @BeforeMethod
    public void login() throws IOException {
        Reporter.log("login",true);
        driver.get("https://demo.actitime.com/login.do");
        FileLib f = new FileLib();
        driver.findElement(By.xpath("//input[@type
='text']")).sendKeys(f.getPropertyData("username"));

driver.findElement(By.xpath("//input[@type='password']")).sendKeys(f.getPropert
yData("password"));
        driver.findElement(By.xpath("//div[text() ='Login ' ]")).click();
    }
    @AfterMethod
    public void logot() {
        Reporter.log("logout",true);
        //driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    }
}
```

```

        driver.findElement(By.id("logoutLink")).click();
    }
}

```

## **Listeners**

\* Listener is a feature available in testng which is used to monitor test execution in runtime and perform appropriate action or operation whenever the test case is getting failed or passed or skipped etc.

\* Whenever we want to implement the listener feature, we should create a class which implements ITestListener and provide implementation (override) to all the methods present in ITestListener.

```

public class ListenerImplementation extends BaseClass implements
ITestListener {

```

```

    @Override
    public void onTestStart(ITestResult result) {
    }

```

```

    @Override
    public void onTestSuccess(ITestResult result) {
    }

```

```

    @Override
    public void onTestFailure(ITestResult result) {
        String test = result.getName();
        TakesScreenshot t = (TakesScreenshot) driver;
        File src = t.getScreenshotAs(OutputType.FILE);
        File dest= new File("./Screenshot/"+test+".png");
        try {
            FileUtils.copyFile(src, dest);
        }
        catch(IOException e) {
        }
    }
}

```

```

@Override
public void onTestSkipped(ITestResult result) {
}

@Override
public void onTestFailedButWithinSuccessPercentage(ITestResult result) {
}

@Override
public void onTestFailedWithTimeout(ITestResult result) {
}

@Override
public void onStart(ITestContext context) {
}

@Override
public void onFinish(ITestContext context) {
}
}

```

\* ITestListener is an interface which is used to receive the failure events from @Listener annotation.

\* @Listner annotation should be declared before the class definition block in every test class which is used to monitor the test execution in runtime and generate the event if any test case is getting failed (else those test cases will not be monitored).

Example

```

@Listeners(com.sateesh.generic.ListenerImplementation.class)
public class CustomerModule extends BaseClass {
    @Test
    public void createCustomer() {
        Reporter.log("createCustomer",true);
        Assert.fail();
    }
}

```

## Group Execution

- \* Executing the collection of similar test scripts is called group execution.
- \* In automation regression suite will be divided into two types
  1. Smoke test scripts
  2. Regression test scripts
- \* In order to achieve smoke testing we go for group execution of testing.
- \* In order to achieve group execution every test script should have a group name along with @Test, one script can have multiple group names.  
Example: @Test(groups="RegressionTest")  
          @Test(groups={"RegressionTest", "SmokeTest"})
- \* We should declare a group key within the testng.xml file.
- \* Group key should be declared before the test tag and after the suite tag (as shown below).

programm

In case of group execution every configuration method should have group names otherwise those configuration methods or annotations will not be executed or will not participate in the group execution.

Example: @BeforeMethod(groups={"RegressionTest", "SmokeTest"})

## Q: How do you perform regional regression testing or unit regression testing in selenium ?

- Regional regression testing means executing changes made and impacted area, whereas in case of unit regression testing we execute only the changes.
- In order to perform unit regression or regional regression testing we use a method and include a tag inside the class tagged as shown below.

```
<suite name="Suite">  
  <test thread-count="5" name="Test">  
    <classes>  
      <class name="com.actitime.testscript.ProjectModule">
```

```

    <methods>
        <include name="createProject"/>
    </methods>
</class>
</classes>
</test>
</suite>

```

**Q: What is runtime polymorphism and where have you used it in selenium and why ?**

```

public class RunTimePolymorphism {
static{
    System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    System.setProperty("webdriver.gecko.driver", "./driver/geckodriver.exe");
}
public static WebDriver driver;
@Test
public void testRunTime() {
    System.out.println("enter the browser name");
    Scanner sc = new Scanner(System.in);
    String browser = sc.nextLine();
    if(browser.equals("chrome")) {
        driver =new ChromeDriver();
    }
    else if(browser.equals("firefox")) {
        driver = new FirefoxDriver();
    }
    driver.get("https://www.google.com");
    String title = driver.getTitle();
    System.out.println(title);
}
}

```

**Parallel execution**

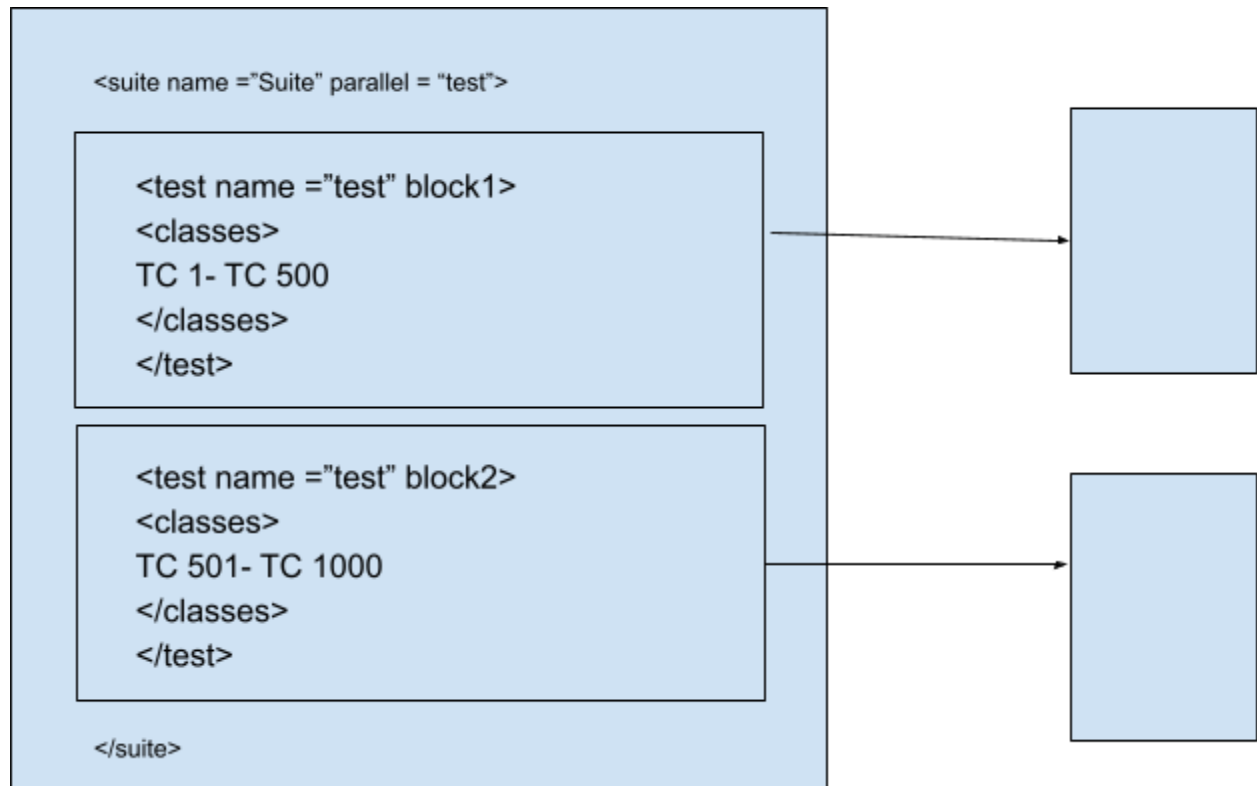
- Executing the multiple test scripts with multiple browsers concurrently or at the same time is called Parallel execution.



- TestNG provides 2 types of parallel execution
  1. Distributed parallel execution.
  2. Cross browser testing or compatibility parallel testing.

### Distributed parallel execution

In order to get the result in early stages we should go for distributed parallel execution.



- If we execute all the 1000 test cases sequentially it may take 20 hours, in order to get the result in early stages we should customize the testng.xml suite file as shown in the above diagram.
- In order to achieve distributed parallel execution, create multiple test blocks and distribute the test scripts and enable the attribute parallel = "tests" in the suite tag.

```
<suite name="Suite" parallel="tests">  
  <test thread-count="5" name="Test1">
```

```

<classes>
  <class name="com.actitime.testscript.ProjectModule"/>
</classes>
</test>

```

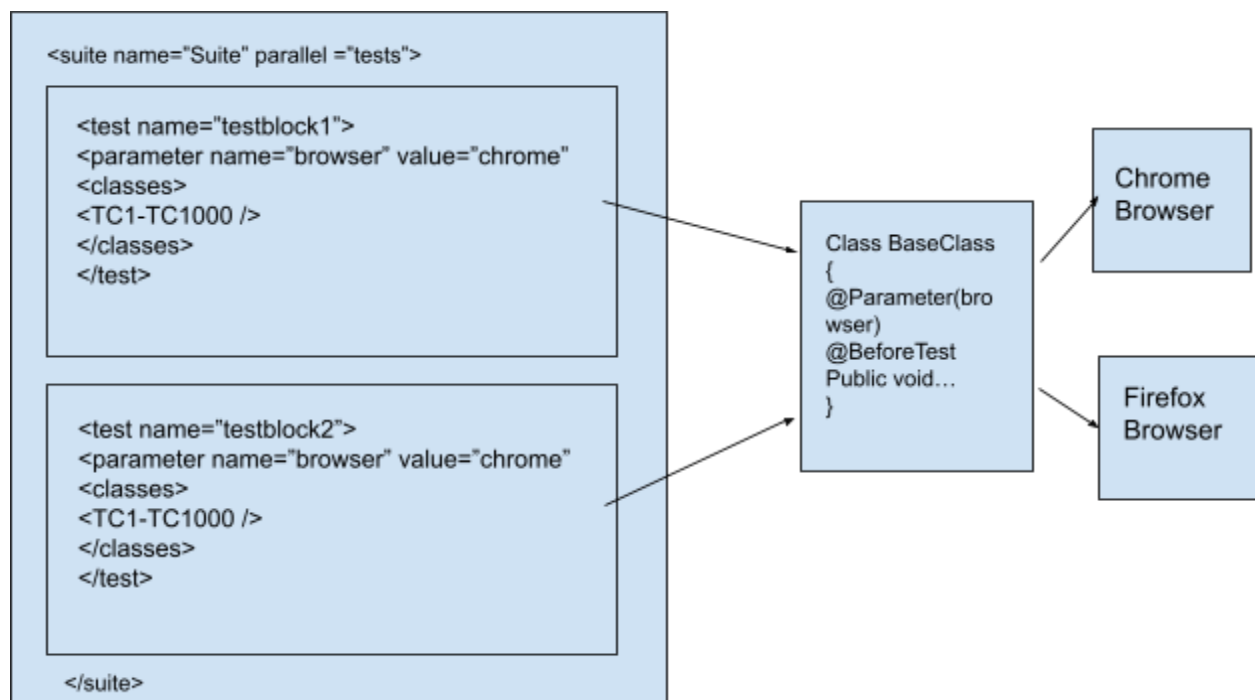
```

<test thread-count="5" name="Test2">
  <classes>
    <class name="com.actitime.testscript.CustomerModule"/>
  </classes>
</test>
</suite>

```

## Cross browser testing

Executing the test scripts in different browsers parallelly at the same time is called cross browser testing.



- In order to achieve cross browser testing we should create multiple test blocks and provide browser parameters for each test block as shown in the above diagram.

- So that when we execute the suite file each test block opens the specific browser using threads and executes all the test scripts in different browsers at the same time parallelly.
- In case of cross browser testing we should mandatorily use `@Parameter` annotation along with `@BeforeTest` and `@AfterTest` annotations in the BaseClass.
- `@Parameters` annotation will be used to receive browser parameters from the xml suite file into the BaseClass or test script.
- In real time to perform cross-browser or cross platform testing we go for Selenium Grid.
- Selenium grid is another tool available in the selenium community which is used to run in another virtual machine.

**\*Q: What is the difference between BeforeTest and BeforeMethod ?**

<b>@BeforeTest</b>	<b>@BeforeMethod</b>
1. BeforeTest will be executed before the execution of the test block present in the xml suite file.	1. BeforeMethod will be executed before the execution of every <code>@Test</code> method(test case).

**@BeforeSuite**

`@BeforeSuite` will be executed before the suite tag present in the testng.xml file or suite file. In real time it is used to connect to the database if the project requires DB connection.

**@AfterSuite**

It will be executed after the suite tag present in the testng.xml file and it is used to close the DB connection in real time.

**@BeforeTest**

It will be executed before the test block present in the testng.xml file and it is mandatory to use in case of parallel execution.

**@AfterTest**

`AfterTest` will be executed after the Test block in the suite file.

## **Encapsulation :**

Hiding the data and binding with methods in order to hide the internal implementation is called encapsulation.

- We achieve this by making the variable private and access outside the class with the help of getters and setters method.

```
public class A {
    private int i;          //declaration
    public A() {
        i=10;              // initialization
    }
    public int getValue() {    // giving only read access
        return i;
    }
    public void setValue(int j) {    //giving write access
        i=j;
    }
}

public class B {
    public static void main(String[] args) {
        A a1= new A();
        int x=a1.getValue();        //utilization
        System.out.println(x);
        a1.setValue(52);            //utilization
        System.out.println(a1.getValue());
    }
}
```

Data will be stored in a variable, in java for any given variable we should perform following steps

1. Declaration
2. Initialization
3. Utilization

There are 2 classes in the above example, the purpose of class A is to only manage the variable i whereas the purpose of class B is only to execute the code.

### Using Encapsulation in selenium

Selenium code to enter admin in the username textbox:

```
driver.findElement(By.id("username")).sendKeys("admin");
```

Or above code can also be written as

```
WebElement untbx = driver.findElement(By.id("username"));
untbx.sendKeys(un);
```

Or

```
WebElement untbx;           // declaration
untbx= driver.findElement(By.id("username")); //initialization
untbx.sendKeys(un);         //utilization
```

**13.10.2021**

### Prog 1.1:

```
package qsp;
```

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

```
public class LoginPage {
    private WebElement untbx;           //declaration
    public LoginPage(WebDriver driver) {
        untbx=driver.findElement(By.id("username")); //initialisation
    }
    public void setUser(String un)
```

```

        {
            untbx.sendKeys(un);
        }
    }
}

```

Prog 1.2:

```
package qsp;
```

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
```

```
public class ValidLogin {
    static
    {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    @Test
    public void testValidLogin() {
        WebDriver driver=new ChromeDriver();
        driver.get("https://demo.actitime.com/");
        LoginPage l=new LoginPage(driver);
        l.setUser("admin");
    }
}

```

Prog 2.1:

```
package qsp;
```

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

```
public class LoginPage {
    private WebElement untbx;    //declaration
    private WebElement pwtbx;
```

```

private WebElement lgbtn;
public LoginPage(WebDriver driver) {
    untbx=driver.findElement(By.id("username"));    //initialisation
    pwtbx=driver.findElement(By.name("pwd"));
    lgbtn=driver.findElement(By.xpath("//div[.='Login ']"));
}
public void setUser(String un)
{
    untbx.sendKeys(un);    //utilisation
}
public void setPassword(String pw)
{
    pwtbx.sendKeys(pw);
}
public void setLogin()
{
    lgbtn.click();
}
//      (OR)
//business logic method
public void setLogin(String un,String pw)
{
    untbx.sendKeys(un);
    pwtbx.sendKeys(pw);
    lgbtn.click();
}
}

```

Prog 2.2:

```
package qsp;
```

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
```

```
public class ValidLogin {
    static
```

```

    {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    @Test
    public void testValidLogin() {
        WebDriver driver=new ChromeDriver();
        driver.get("https://demo.actitime.com/");
        LoginPage l=new LoginPage(driver);
        l.setLogin("admin","manager");
    }
}

```

**-Selenium code to enter valid login and invalid login is(or setLogin() method can be used to enter valid and invalid input):-**

```

package qsp;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;

public class ValidLogin {
    static
    {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    @Test
    public void testValidLogin() throws InterruptedException {
        WebDriver driver=new ChromeDriver();
        driver.get("https://demo.actitime.com/");
        LoginPage l=new LoginPage(driver);
        l.setLogin("admin1","manager1");
        Thread.sleep(3000);
        l.setLogin("admin","manager");
    }
}

```



```
}  
}
```

-When we execute the above code we may get

**StaleElementReferenceException**, because when it clicks on login button after entering invalid username and password, page will be reloaded and address of the element will be changed. But the reference variable such as untbx, pwtbx and lgbtn will be holding the old address. It will try to enter valid username and password using old address which is no longer exist(invalid), hence we get the exception.

### **-Script to explain StaleElementReferenceException:-**

```
package com.actitime.testscript;
```

```
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.testng.annotations.Test;
```

```
public class DemoPOM {  
    static  
    {  
        System.setProperty("webdriver.chrome.driver",  
"./driver/chromedriver.exe");  
    }  
    @Test  
    public void testUserName() {  
        WebDriver driver=new ChromeDriver();  
        driver.get("https://demo.actitime.com/");  
        //stores the username textbox address as @a1 in untbx  
        WebElement untbx=driver.findElement(By.id("username"));  
        //refresh the page and address of the username textbox will change  
to @p1  
        driver.navigate().refresh();  
        //try to enter admin in username textbox using old address i.e, @a1
```

```
        untbx.sendKeys("admin");
    }
}
```

## **PAGE OBJECT MODEL:**

-Page Object Model is one of the java design pattern which is used to store the objects(elements). POM concept is used by the developers to develop the web page and it is also used by the automation engineers to test the web pages.

-It is also used to avoid StaleElementReferenceException.

-In POM, we declare the element by using @FindBy annotation and we initialise the element by using PageFactory.initElements() method.

-@FindBy annotation should be imported from org.openqa.selenium.support package.

-The syntax is:-

Syntax 1:-

```
@FindBy(Locatorname="Locator Value")
private WebElement ElementName;
```

Syntax 2:-(for multiple elements)

```
@FindBy(Locatorname="Locator Value")
private List<WebElement> ElementName;
```

-PageFactory.initElements() method will take 2 arguments(parameters)

1. WebDriver(driver)
2. Object of POM class or page class(this)

### **Q: How POM will avoid StaleElementReferenceException?**

A: initElements() method will only load the element(only declares), but it will not initialise actually. Elements are actually initialised during the run time when we try to perform action on the elements. This process is called as lazy initialisation. This will avoid StaleElementReferenceException.

### **Note**

The class in which elements of the web page are stored by using @FindBy annotation is called POM class or Page class.

Example : Login page, homepage, task list page etc.

The class which we execute by using @Test (test scripts are written) is called a test class.

```
package com.actitime.pom;
```

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

```
public class LoginPage {  
    @FindBy(id="username")  
    private WebElement untbx;           //declaration  
  
    @FindBy(name="pwd")  
    private WebElement pwtbx;  
  
    @FindBy(xpath="//div[.='Login ']")  
    private WebElement lgbtn;  
  
    public LoginPage(WebDriver driver) {  
        PageFactory.initElements(driver, this); //initialization  
    }  
    public void setLogin(String un, String pw) {  
        untbx.sendKeys(un);  
// utilization  
        pwtbx.sendKeys(pw);  
        lgbtn.click();  
    }  
}
```

```
-----  
package com.actitime.testscript;
```

```
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.openqa.selenium.support.PageFactory;
```

```

import org.testng.annotations.Test;

import qsp.LoginPage;

public class TestClass {
    static
    {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    @Test
    public void testValidLogin() {
        WebDriver driver=new ChromeDriver();
        driver.get("https://demo.actitime.com/");
        LoginPage l=new LoginPage(driver);
        PageFactory.initElements(driver, this);
        l.setLogin("admin","manager");
    }
}

```

**Q: What happens if we don't use initElements() in POM class or anywhere else ?**

We get NullPointerException

**Q: Can we develop a POM class without a constructor?**

Yes

**Q: Can we use initElements() in test or method class ?**

Yes, but we should explicitly call initElements() in test class (or main method class).

**Q: How will you declare an element in POM?**

By using @FindBy annotation.

**Q: How do you initialize an element in POM ?**

By using initElements() of the PageFactory class.

**Q: What is an object repository ?**

It is a location where we store the objects or elements. POM is also called as Object Repository or Page Object Repository

**Q: What is the difference b/w Page Object Model and Page Factory ?**

POM	Page Factory
It is a java design pattern/ concept.	It is a class which implements the POM concept.

**Advantages of POM**

- 1.Maintenance of WebElements is easy, since webelements are maintained based on the page.
2. Modification of webelement locators is easy whenever the UI is changed.
3. We can avoid StaleElementReferenceException.
4. Xpath and locators of the webelements are reusable, so that no need to put the same effort to rewrite the xpath again.
5. Object repositories can be easily shared across the team members via Github.

**Q: What is page class or POM class?**

Page class is a class in which elements are declared by using @FindBy annotation.

**\*\*\*Note**

In real time the number of POM classes will be equal to the number of web pages present in the application.

**Q: How do you handle multiple elements in POM ?**

```
package com.actitime.pom;
```

```
import java.util.List;
```

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

```

public class CheckBoxPage {

    @FindBy(xpath="//input[starts-with(@type,'checkbox')]")
    private List<WebElement> boxes;

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

    public void setCheckBoxClick() {
        for(int i =0;i<boxes.size();i++) {
            boxes.get(i).click();
        }
    }
}

```

-----

```

package qsp;

```

```

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;

```

```

import com.actitime.pom.CheckBoxPage;

```

```

public class AllCheckBox {
    static {
        System.setProperty("webdriver.chrome.driver",
            "./driver/chromedriver.exe");
    }
}

```

```

@Test
    public void testCheckBox() {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/checkbox.html");
        CheckBoxPage c= new CheckBoxPage(driver);
        c.setCheckBoxClick();
    }
}

```

}

### **Challenges faced in selenium:**

1. We can automate only web applications.
2. We can take the screenshot only in png format (other formats not supported).
3. We cannot minimize the browser.
4. We cannot specify the password in encrypted format.
5. We cannot take the screenshot of the popup.
6. We cannot handle file download popup of firefox browser in selenium.
7. We cannot perform action on the browser which is already opened.
8. We cannot compare the images (or image testing).
9. We can't automate the test cases which contain captcha, OTP, barcode scanning, biometric scanning etc.
10. We can't automate non functional test cases (like performance testing) for that we have to integrate J-meter with selenium.
11. We cannot take a screenshot of a particular location of the browser.

### **Debugging or How to debug in eclipse**

In eclipse it provides 2 types of execution mode

1. Run mode: we can execute the script without pause in the eclipse.
  2. Debug mode: it will execute the script based on the user instruction and it pause the execution whenever break point available inside the script.
- in real time debug mode will be very helpful to know the exact issue of the test script wherever the test is getting failed.

### **Debugging feature available in eclipse**

1. **Break point (ctrl+shift+b)**: it is used to pause the execution in the runtime whenever the test script executes in debug mode.
2. **Step over (f6)**: execute the test script line by line.
3. **Step info (f5)**: execution control goes inside the method and pauses the execution in the 1st line of the method (press f6 for line by line execution).
4. **Resume (f8)**: execute the test script from one breakpoint to another breakpoint.
5. **Step return (f7)**: execution control comes out of the called function (method) by executing all the statements in pause mode.

### Framework:

- Framework is a set of rules and guidelines or best practices to be followed while automating any application.

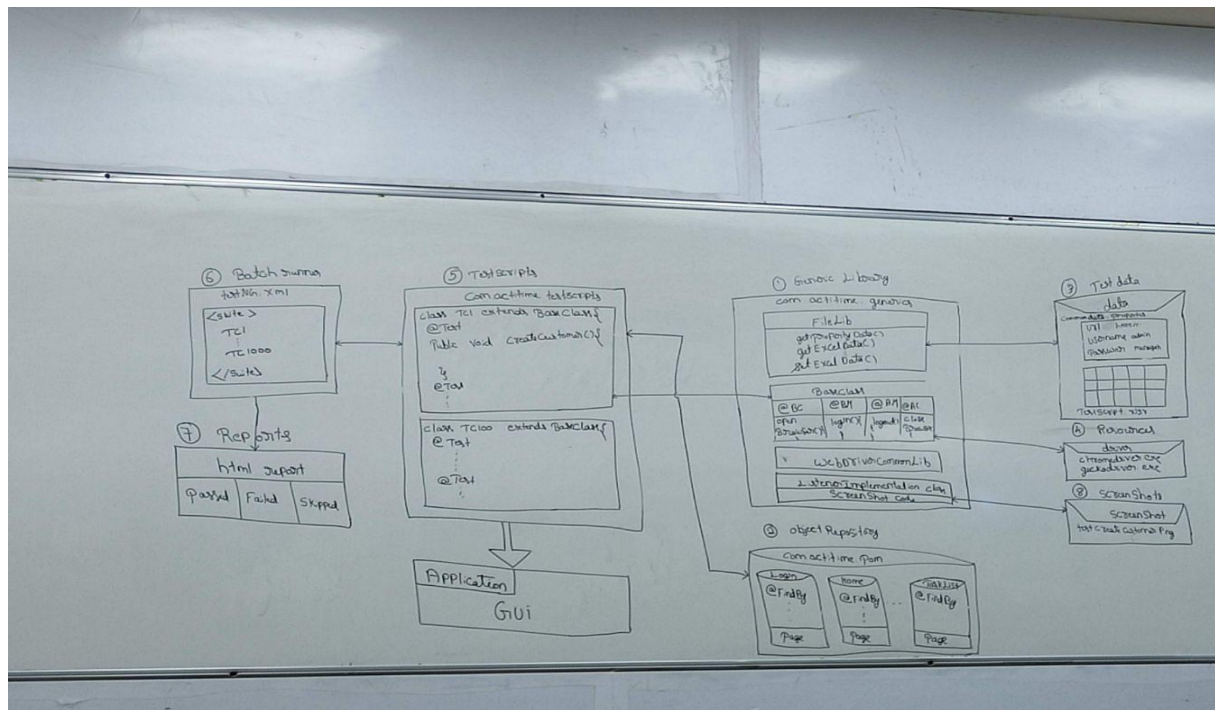
Or

- Framework is a collection of reusable components that means automation test script development, execution and modification to be easier and faster.

Or

- Framework is a well organized structure of reusable components where one driver script (testng.xml) will take care of entire batch execution without any manual intervention.

## Architecture of Hybrid Framework



- Framework is a set of rules and guidelines or best practices to be followed while automating any application.
- Inorder to execute the test script with multiple inputs or test data, we use an excel and property file, so we call our framework a Data driven framework.
- Inorder to avoid writing repetitive steps again and again we use lots of reusable methods, so we call our framework a method driven framework.



- Since we maintain our framework module wise, we also call our framework a Modular driven framework.
- Since it is a combination of 2-3 frameworks hence it is called a Hybrid driven framework.
- In the beginning of the execution first it executes the BaseClass which is present in the Generic package which contains all the configuration methods like @BeforeClass, @BeforeMethod, @AfterMethod and @AfterClass. First it executes the @BeforeClass which contains the code for opening the browser, then it will execute the Login code which is present in @Beforemethod.
- Then it will executes the @Test method where actual test scripts are written in test script package, while executing the test scripts it will take the test data from the excel file with the help of Apache POI jars and performs action on the GUI application by calling necessary methods present in the POM class.
- Once it executes the test method it will execute the Logout code which is present in @AfterMethod.
- Like this it will execute all the test cases one after the other with help of Batch runner(testng.xml).
- After the execution of all the test cases it will close the browser which is present in @AfterClass.
- Since we are using TestNG it will automatically generate the default HTML report (test-output folder), which contains number of test cases passed, failed and skipped.
- Since we have implemented the Listener feature of testng it will automatically take the screenshot of failed test cases in the screenshot folder.

## **Framework Folder Structure:**

Actitime

```
|____Src
|    |____com.actitime.generic
|    |    |____BaseClass
|    |    |____FileLib
|    |    |____ListenerImplementation
|    |____com.actitime.pom
|    |____LoginPage
|    |____HomePage
|    |____TaskListPage
|    |____com.actitime.testscript
|    |    |____CustomerModule
|    |    |____ProjectModule
|____data
|    |____commondata.preproperty
|    |____testscript.xlsx
|____screenshot
|____test-output
|____testng.xml
```

## **Source control tool (configuration management tool or Version control tool)**

1. Github
2. SVM (SubVersion)
3. VSS (Visual Source Safe)
4. Perforce

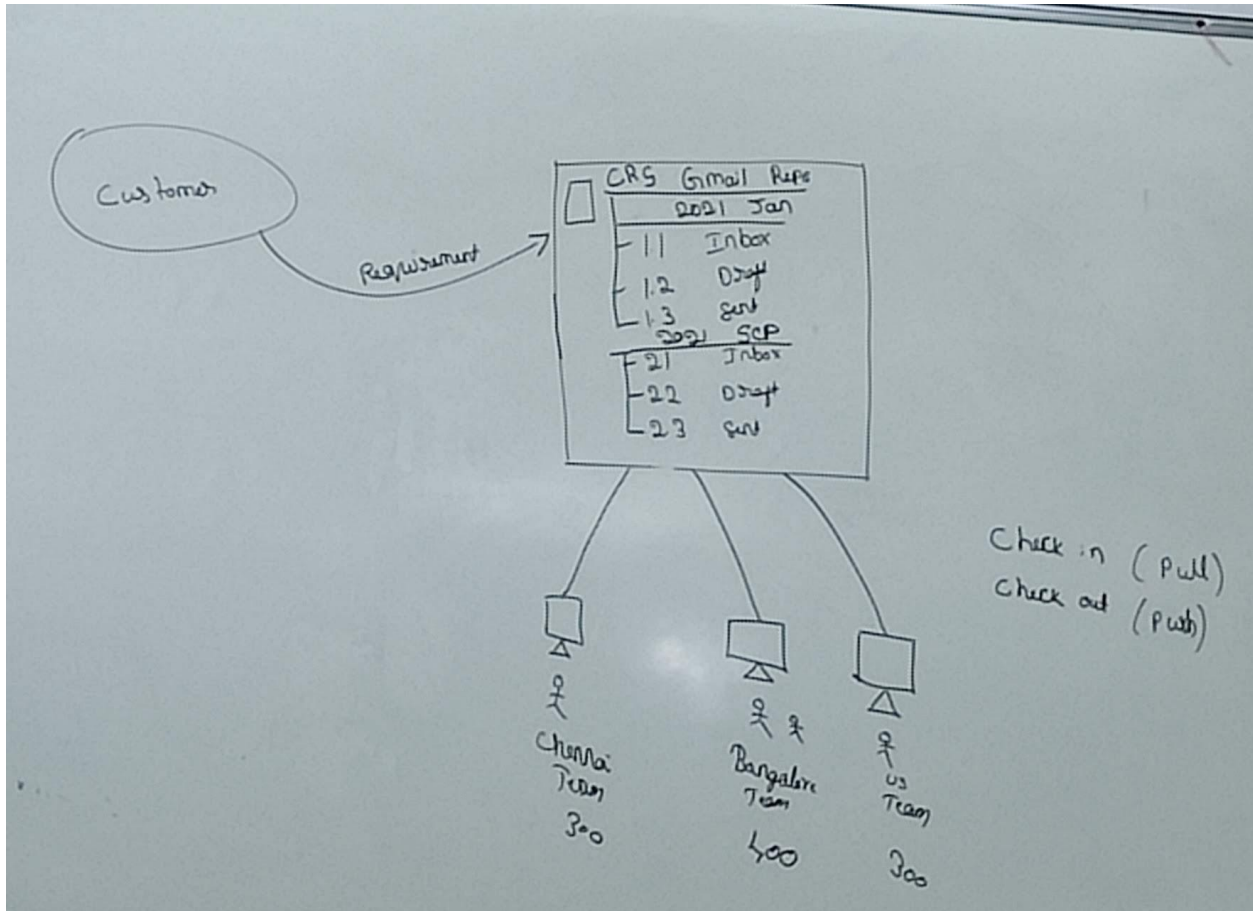
### **Github**

Github is a decentralized online cloud repository where we can maintain our entire framework in one place, so that it can be shared easily with multiple engineers working on the same project in different locations.

### **Advantages of Github**

1. Decentralized repository.
2. Cloud repository.
3. No need for a system engineer to maintain Github.

4. File sharing between the team members will be easier.
5. Master copy of the framework available in the github so that we can perform batch execution any time.



There are two types of server available in github

1. Git Server (github)
2. Git Client
  - a. Git bash
  - b. Git command prompt
  - c. Git GUI
  - d. Git Desktop
  - e. EGit

Github repository URL: <https://github.com/Sateesh141996/SeleniumProject.git>

Github token: [ghp\\_A7T7qp3QPeneaaEXVcFeCuaDDNY6IG0HqQp6](#)

## Experience notes

=====

### Xpath axes

Xpath axes are the special keywords of xpath which has following syntax

`//axesName :: tag`

example - `/decendent::a` ---> means `//a`

**Axes names:**

1. ancestor
2. decedent
3. child
4. parent
5. following-sibling
6. preceding-sibling

table

  |\_Tr

1 |\_\_td        2 java

2 |\_\_td        1

3 |\_\_td

4 |\_\_td

5 |\_\_td        selenium

    1|\_\_a ide

    2|\_\_a webdriver

For td[3] - td[1],td[2] are preceding-siblings and td[4].td[5] are following-siblings

//td[3]/preceding-sibling::td[2] → java

//td[3]/following-sibling::td[2] → selenium

//td[3]/parent::tr

//td[3]/child::a[2]

//td[3]/ancestor::table

**Xpath with starts-with function**

We can use the start-with function on the xpath either by using text() or attribute in it. The syntax is as followed

//tag[starts-with(text(),'value')]

//tag[starts-with(@AN ,'AV')]

Html code of an element is: <a>Inbox(7)</a>

Xpath is : //a[starts-with(.,'Inb')]

Example Html code-

```
<table border ="1">
<tr>
<td>selenium</td>
<td>100</td>
</tr>
<tr>
<td>selenium</td>
<td>300</td>
</tr>
</table>
```

```
Html
|__body
    |__table
        |__tbody
            1 |__tr
                1 |__td java
                2 |__td 100
            2 |__tr
                1 |__td selenium
                2 |__td 300
```

Xpath for selenium price

//td[.='Selenium']/following-sibling::td

### Assignment

**Q: WAS to print the content of the webtable present in any given website or URL should be taken from the user.**

```
public class ApexTableContent {
    static {
```

```

        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/Demo.html");
        List<WebElement> book =
driver.findElements(By.xpath("//td[starts-with(.,'sel')]"));
        List<WebElement> price =
driver.findElements(By.xpath("//td[starts-with(.,'sel')]/following-sibling::td"));
        int count = book.size();
        for(int i=0;i<count;i++) {
            String bookname = book.get(i).getText();
            String bookprice = price.get(i).getText();

System.out.println("BookName:"+bookname+"-->"+"BookPrice:"+bookprice);
        }
        driver.close();
    }
}

```

**Q: WAS to click on all the checkboxes present in the given web app or URL should be taken from the user.**

Html code:

```

c1: <input type="checkbox" /><br>
c2: <input type="checkbox" /><br>
c3: <input type="checkbox" /><br>
c4: <input type="checkbox" /><br>
c5: <input type="checkbox" /><br>
c6: <input type="checkbox" /><br>
c7: <input type="checkbox" />

```

```

public class ApexCheckBox {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/SATEESH/Desktop/checkbox.html");
        List<WebElement> boxes =
driver.findElements(By.xpath("//input[starts-with(@type,'checkbox')]"));
        int count = boxes.size();
        for(int i =0;i<count;i++) {
            boxes.get(i).click();
            Thread.sleep(1000);
        }
        for(int i=count-1;i>=0;i--) {
            boxes.get(i).click();
            Thread.sleep(1000);
        }
        driver.close();
    }
}

```

**Q: WAS to delete the text present in the email textbox of OpenSourceBilling app character by character.**

**Hint: by using backspace**

```

public class ApexOpenSourceBilling {
    static {
        System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    }
    public static void main(String[] args) throws InterruptedException {
        WebDriver driver =new ChromeDriver();
        driver.get("http://demo.opensourcebilling.org/en/users/sign_in");
    }
}

```



```
        String value =  
driver.findElement(By.xpath("//input[starts-with(@id,'password')]")).getAttribute("v  
alue");  
        int count = value.length();  
        for(int i=count-1;i>=0;i--) {  
  
driver.findElement(By.xpath("//input[starts-with(@id,'password')]")).sendKeys(Key  
s.BACK_SPACE);  
                Thread.sleep(1000);  
        }  
        driver.close();  
    }  
}
```



## **Exceptions**

=====

### 1. IllegalStateException (selenium/unchecked)

We get this exception whenever the\_path of the driver executable file is not set.

### 2. InterruptedException (java/ checked)

We get this exception whenever we use Thread.sleep()

### 3. NoSuchElementException (selenium/ unchecked)

We get this exception whenever locators are not matching with any of the elements.

#### 4. InvalidSelectorException (selenium/unchecked)

We get this exception whenever there is a syntax error or syntax mistake  
In `cssSelector()` or `xpath()`

#### 5. TimeoutException (selenium/unchecked)

We get this exception whenever locators are not matching or whenever elements are not within the specific time given in the `explicitWait`.

#### 6. UnsupportedOperationException (selenium/unchecked)

We get this exception whenever we use `deSelect()` on a single select listbox.

#### 7. UnexpectedTagNameException (selenium/unchecked)

We get this exception whenever we try to pass `WebElement` to the `select` class which is not a type of listbox.

#### 8. NoAlertPresentException (selenium/unchecked)

We get this exception whenever we try to handle the alert popup when the popup itself is not present.

#### 9. UnhandledAlertException (selenium/unchecked)

We get this exception whenever we try to perform action on the browser without handling alert popup.

#### 10. IndexOutOfBoundsException (java/unchecked)

We get this exception whenever we pass invalid index value inside the for loop.

#### 11. WebDriverException (selenium/unchecked)

We get this exception whenever we pass the relative path of the file to the `sendKeys()`.

#### 12. InvalidArgumentException (selenium/unchecked)

We get this exception whenever we pass the wrong path of the file.

13. AWTException (java/checked)

We get this exception whenever we use Robot class.

14. IOException (java/checked)

We get this exception whenever we use a runtime class or whenever we try to open any files.

15. NoSuchWindowException (selenium/unchecked)

We get this exception whenever we try to perform an action on the browser which is already closed(invalid).

16. SessionNotCreatedException (selenium/unchecked)

We get this exception whenever the driver executable version is not matching with the browser version.

17. NoSuchFrameException (selenium/unchecked)

We get this exception whenever we pass invalid arguments to the frame().

18. ElementNotInteractableException (selenium/ unchecked)

We get this exception whenever we try to perform action on the disabled elements.

19. JavascriptException (selenium/unchecked)(generic exception)

We get this exception whenever there is a mistake in the syntax of javascript statement.

20. FileNotFoundException (java/checked)

We get this exception whenever we use FileInputStream class.

21. EncryptedDocumentException (apache/checked)

We get this exception whenever we try to read data from the excel file or whenever we use the WorkbookFactory class.

22. TestNGException (TestNG/ unchecked)

We get this exception whenever tests depend on each other.

23. StaleElementReferenceException (selenium/unckecked)

We get this exception whenever the address of the element is old.

=====

## **Resume Building**

### **1. How to get more requirements in Naukri.com?**

Use keywords, update daily

### **\*\*\*2. Tell me about yourself ?**

30 points.

## **Daily work / job of automation engineer**

1. Selecting manual test cases for automation.
  - a. Take test cases from the regression suite(repository).
  - b. Execute the test case manually.
  - c. Check for manual intervention like OTP, captcha, barcode scanning, card swiping, audio detection, image verification etc.
2. Preparing test data.
3. Developing test class.
4. Preparing POM class.
5. Executing the test script.
6. Analyzing the report.
7. Debug the code to fix the errors and execute.
8. Preparing and sending daily status reports.
9. Attending daily standup meetings.
10. Giving a demo to the clients.
11. Developing generic classes.
12. Preparing automation related documents.
13. Enhancing the framework.
14. Giving KT to the new team members.