AUTOMATION PDF -→

1)What is automation testing?

→Converting manual test cases to test script with the help of tool is called automation testing.

2)Why we go for automation testing?

i)To save time.

ii)For better accuracy.

iii)To sustain in market competition.

iv) Less resource we can perform more work.

3) Which test cases we are not going to automate?

- Capture related test cases.
- OTP related test cases.
- Bar code related test cases.
- Gaming test cases

MP3, MP4 related test cases.

If manual test cases are not correct we cannot automate those test cases.

Types of Automation Testing:

1. Functional testing (e.g., GUI, API)
2. Regression testing
3. Unit testing
4. Integration testing
5. System testing
6. Acceptance testing

7. Load testing

8. Performance testing

9. Security testing

Automation Testing Tools:

1. Selenium (Web)

2. Appium (Mobile)

3. TestComplete (GUI)

4. JUnit (Unit testing)

5. TestNG (Unit testing)

6. Cucumber (BDD)

7. Pytest (Python testing)

8. QTP (QuickTest Professional)

9. Rational Functional Tester (RFT)

Automation Testing Frameworks:

1. Data-Driven Framework

2. Keyword-Driven Framework

3. Hybrid Framework

4. Behavior-Driven Development (BDD)

5. Page Object Model (POM)

Q: what is Selenium ?

Selenium is a free ,open source Automation Tool,it can Automate web-Based Applications . but Can't automate Stand-alone and Client-Server applications.

Quick Notes:

Developed by---------- Jason Huggins

Year----------2004

Company (worked in) - ------Thoughtworks, Chicago, USA

Former Name--------- ---------Javascript Test Runner

Renamed As----------Selenium

Reason (for renaming)------- To Compete with HP Mercury Automation Tool.

Selenium Versions / Components / Flavours:

1.Selenium Core

2. Selenium IDE (Integrated Development Environment)Record and Playback tool.

3.Selenium RC (Remote Control), also Called as Selenium 1.0.

4. Selenium Webdriver (2007) / Selenium 2.0.

Current Stable Version -------3.141.59

Upcoming (Alpha)----------4.0

5. Selenium Grid

Note: Selenium -----Selenium Commands which are Used in Selenium IDE

-----> Selenium Supports all programming languages - Java, C#, Javascript, Perl, Ruby, Python, R, TCL, Elixir, Haskell.

---->Selenium Supports all the browsers:--Google Chrome, Firefox, Opera, Safari.

----->Selemium Supports all the Operating Systems: Windows, Mac, Linux, except for Unix

Types of Applications:

1. Web based Applications:-- A web based appli- Cation is any program that is accessed over a network Connection using HTTP, rather than existing within a devices memory. web browser. Web based applications.it ofter web based applications are also known as web apps.

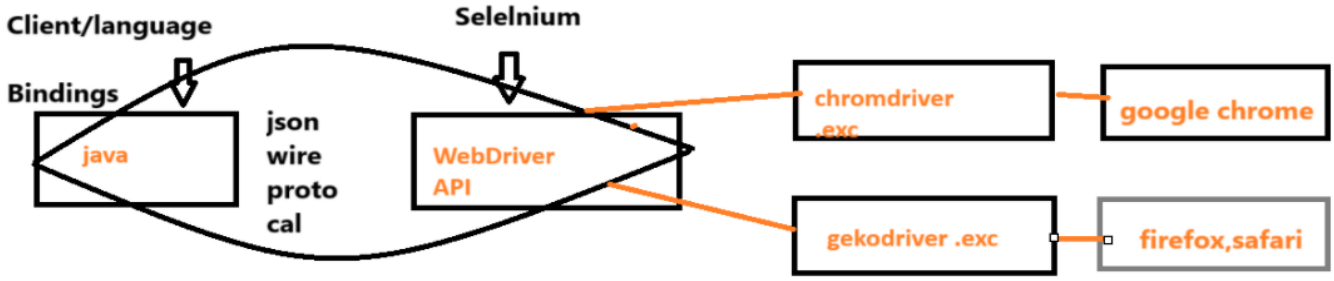
2. Client Server Applications:

It Consists of a client program that Can- Sumes Services provided by a Server program .The Client reqests Services from the Server by Calling functions in the Server application.

3. Stand Alone Applications:---

It is an application that runs locally on the device and dosent require anything functional. All the logic is buit in to the app, so it doesn't need an internet connection are any other sevices installed . These type of applications are not bound to any specific platform.

Java----Selenium----ARCHITECTURE:



JSON --------> Javascript object Notation

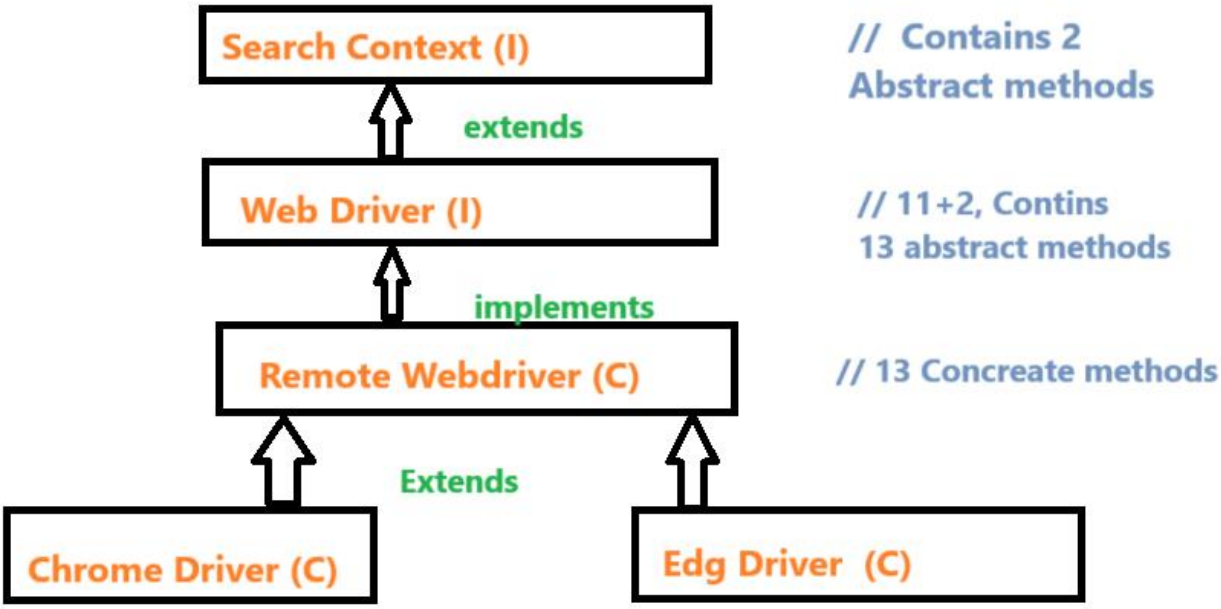
API ---------> Application programming Interface.

➔ Selenium webdriver API Supports all the programming languages and it Communicates With language bindings like java by using JSON Wire protocol.

→ These two Combined to communicate with the browsers by using respect

tive driver files like executable Chrome driver.exe, geckodriver.exe etc.

WEB DRIVER ARCHITECTURE :



Explanation:

Search Context is the Super most interface, Web Driver interface extends itand all the 13 abstract methods of webdriver are given implementation in Remote WebDriver class all those Concrete methods are overriden in respective browser Classes like , chrome Driver class Firefox Driver Class etc.

=======> As per Selenium Standards, we always upcast browser Classes to web Driver interface

1. Generalization (notspecifying any type)

2. Runtime Polymorphism (At runtime, we Can decide in Which browser my code will run) .

3. To get all the 13 methods required for Automation Testing.

Explain the following statement:

WebDriver driver=new ChromeDriver();

i. WebDriver is a interface

ii. driver is reference variable

iii. = is assignment operator

iv. new is keyword

V. ChromeDriver is constructor

vi. ; is statement delimiter

WEBDRIVER ABSTRACT METHODS :

(Browser window Related methods)

1.Get ()

2.NaviGate () navigate have four contains:

1. TO() 2.forword() 3.backword () 4.refresh ()

3.GetCurrentUrl ()

4.GetPageSource ()

5.GetTitle ()

6.GetWindowhandle ()

7.GetWindowhandles ()

8.Manage ()

9.SwitchTo ()

10.FindElement ()

11.findElements ()

12.close ()

13.quit ()

WebDriver methods : It is interface which have 13 abstract methods which are useful for Automation proces.

1.get ():

It is used to open application/it is used to enter url.

Ex:-

WD d = new CD ();

driver.get ("url");

Note: if url is not proper then we will get webdriver exception

2.Navigate(): It is also used to open the application, and

move forward, Back word and refresh the browser.

Note: It is alternative method for get().

Ex: WD d = new CD();

drives. navigate().to ("google");

Thread.sleep (3000);

driver. navigate().to ("amazon");

drives. navigate().back();

dsines. navigate (). forward();

drives-navigate().Refresh();

Thread sleep (2000);

}

}

Thread.sleep(): It is type of wait which comes from Java.

(1)It will wait for particular interval of time and then perform the next scenario.

(2)Here time is always given in milliseconds.

(3)Thread.sleep(millisecond);

Difference blw get () and Navigate ()

get ()	navigate ()
1)used to enter url (only one work) 2)it will not show browser history 3)get will wait until complete page loaded	1)used to enter url,BackWord ,forword and refresh (4Works) 2)it will show browser history 3)it will not wait until complete page is loaded.

3)close ():-it is used to close the tab. (it will close current tab)

Syntax :-driver.close ():

4) Quit ():-alternative method for close. (It will close all tabs)

Syntax :-driver.quit ();

maximize():- It is used to maximize the Browser.

Syntax:- driver. manage(). Windows (). maximize();

Minimize ():-It is used to minimize the Browses.

Syntax:- driver. Manage (). window(). minimize();

Setsize(): It is used to change the size of Browser. Which accepte Dimenssion clan arguments.

Ex:-

```
WebDriver driver=new EdgeDriver();
      driver.get("URL");

   Dimension d = new Dimension(100, 200);
  driver.manage().window().setSize(d);
```

HEIGHT WIDTH

Parameters

setposition: - It is used to chage the position of browse.r which accepts point clan argument.

Ex:-

```
    WebDriver driver=new EdgeDriver();
     driver.get("URL");

    Point p =new Point(100, 200);
    driver.manage().window().setPosition(p);
}
```

5) getTitle():-

→with the help of this method we can fetch the page.

Syntax: - Type 1) :- String title=driver.GetTitle();

System.***out***.println(title);

Type 2):- System.***out***.println(driver.GetTitle(););

6) getCurrentUrl():-

→with the help of this method we can fetch the url of a web page.

Syntax:- Type 1):- String Url = driver.getCurrentUrl();

System.***out***.println(Url);

Type 2):-

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

7) getPageSource():-

→To fetch the frontend source code of a webpage we have to use this method.

Syntax:- Type 1):- String source = driver.getPageSource();

System.***out***.println(source);

Type 2):-

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

8)SwitchTo ():-

→used to Switch over location from webpage to popus,Frames,windows.

Syntax:-

driver.switchTo().newWindow(WindowType.***TAB***);

Locators (heart of the Automation): -

→ There are static methods of abstract By class.

→Locators are used to find one or more elements on webpage.

→ Locators acts as argument for findelement () & findelements ().

Note: findelement () and findelements ()are used to find the elements on webpage.

Difference blw findelement () and findelements ()

Findelement ()	Findelements ()
(1) it will give address of 1st element matching. (2) ReturnType is webElement. (3)if element is not found it will give "No Such Element Exception".	(1) it will give address of all elements matching (2) ReturnType is List <WebElements> (3) if elements is not found it will give " size zero Exception".

Types of Locators:-

(1)TagName (String arg)→ not Unique

(2) ID (String arg)→Unique and Faster one

(3)name (String arg)

(4) Class name (String arg)

(5) LinkText (String arg)

(6) PartialLinkText (arg)

(7) Css Selector (String arg)

(8) Xpath (String arg) slowest

HTML (hypertext markup language:-

→ used to create frontend webpage.

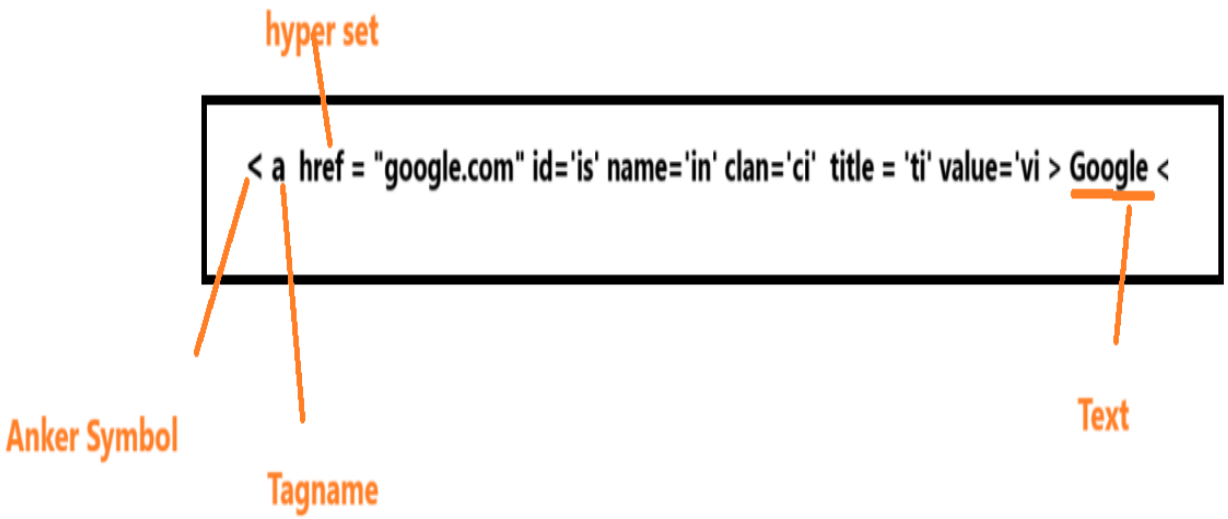
To Design any webpage 3 things are required

(1)Tagname

(2)Attribute

(3)Text

Ex



Rules of HTML:-

(1)Always start with <html> tag and close with </html>.
(2) Inside it you can write <Body> and closeBody </body>

Syntax :- <html>

<Body>

<Body>

</html>

Steps to Inspect :-

(1)Right Click on webapplication and click on inspect
(2) Enable the arrow key (Take Blue visibility)
(3)In web Application, whichever Element you want to find click on that one so that Blue visibility can be highlighted.
(4) Identity the Element like id, name, classname, text Copy the "attribute value".
(5) press control+F check whether element is 1 of 1 or not
(6) If element is 1 of 1 we can paste in our script
(7) In our script we have to write
driver.findElement (By-Locator ("Av"));

(8)Sendkeys(): This method is used to write in our Webapplication.

Locators:-

(1)Tagname:- We can't use Tagname Because it is not Unique But it is faster locator among all locators.

Ex :-//input => It will give more than 1 of 1 matching.

(2)Id:- It is unique and fastest by id we can make matching 1 of 1

Syntax: driver.findElement(By.id("Av"));

(3)name: If id is not there inside the html code. then we can go through name locator.

Syntax: driver.findelement (By.name("Av"));

(4)classname:- if id, name both locators are not there

or if there matching is not 1 of 1 then we can go through class name locator

Syntax: driver.findelement (By.classname("Av"));

(5)Visible Text/ link Text :- If id, name, clansname is not there.

[this Case is very Lase case] then we can go through linktext locator/visible Text.

Syntax: driver.findelement (By.LinkText("Av"));

(6)partial link Text:-

Partial Link Text is a locator strategy in Selenium used to identify links on a web page based on a portion of the link text.

Syntax: driver.findelement (By partiallinktext("Inbox");

(7) Css selector:- for this locator we can take tagname and any one of the attribute we can take.

Rather than id, clans, name any other attributes are there then also it will work.

Syntax: driver.fe (By.cssselector("tagname [AN=’AV’]");

(8)Xpath :- xpath is the path travelled in the HTML tree to find an Element. xpath is one of the locator which covers all Possible ways to find an element.

There are 2-types xpaths :-

(1)Absolute xpath

(2) Relative xpath

Difference b/w Absolute xpath and Relative xpath:-

Absolute xpath	Relative xpath
(1) it is use to navigate frome root of parent to immediate child.	(1) it is used to navigate from root of parent to any child
(2) to achieve absolute xpath we need to use (/) single forward slash.	(2) to achive Relative xpath we need to use (//) double forword slash.
(3) xpatth of Absoulute is to Lengthy	(3) xpath of Relative is short.

Note: xpath of Absotalute is too large hence we can Prefer xpath by Relative always...

Cases of Relative xpath :-

(1)xpath By attribute
(2) xpath By Text
(3)xpath contains
(4)xpath By index

(1) Xpath By Attribute:-

➔ Shortest distance (//)

➔time less

➔Script is short

It is applicable for all attributes

Ex:- id="1234", name=" svg", Placeholder="abc" etc.

formula: - // tagname[@AN = 'AV’]

Case 2:- xpath By Text:- > text <

formula:- //tagname [text() = ‘text’]

Cases 3)Xpath By Contains:-

(1) using Attribute //tagname [contains (@AN, ‘AV’)]

(2) using text //tagname [contains (tent(), 'text')]

Cabe4] xpath By Group Index:-

If we want to convert multiple matching to single matching if 1 of 1 then we have to go through xpath By Group index.

formula: - (xpath Expression) [index]

Questions:→

(1) which is Trustable locator ?

➔Id

(2) which is the fastest locator ?

➔ Tagname

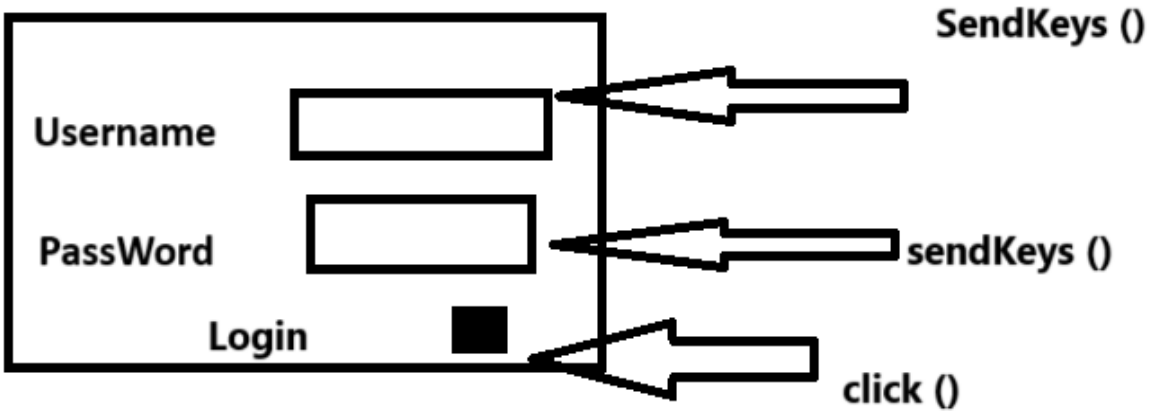
(3) which is slowest locator ?

➔ Xpath

WebElement: It is an interface used to perform action on element present in a browser / webpage.

WebElement methods:-

Ex:-



Since it is an interface all the methods are abstracts methods only.

SendKeys ():-

This method is used to Enter the value in the input field/text field.

Click ():- This method is used to click on Buttons, links, Radio Buttons and checkboxes etc.

Clear ():- It is used to clear value present in the text field / text box.

Ex: driver.get("https://www.facebook.com");

driver.findElement(By.*xpath*("//input[@id='123']")).sendKeys("abc");

driver.findElement(By.*xpath*("//input[@id='123']")).clear();

isEnabled():- It is used to verify Element is enabled or disabled

Returntype of this method is boolean.

Note:- Returntype of all methods which are starting from is are boolean

Ex:- driver.get ("FB.com");

Webelement btn = driver.findelement (By.xpath("//button[@name="lopin']"));

```
if (btn. is enabled()){
sysout ("Element is enabled");
}
else {
 Sysout ("Element is disabled");
}
```

is selected: is selected () is used to verify whether radio button, checkbox is selected or not.

Returntype of isselected method is boolean. it returns either true or false.

isdisplayed():-

This method is used to verity element is present or not. [Returntype is boolean)

getText: -

It is used to get the text which is On webpage its returntype is s String.

List Box /DropDown :-

it is The element from which we can select one option at a time. is called as "ListBox/ DropDown To handle List Box we will be hiving "select class"

Ex:- How to handle ListBox/Dropdown ?

⇒ step 1:- Identity Dropdown which need to handle and Store it in a reference variable.

Step2:- create object of select class which accepts webelement argument.

Ex:- Select s=new select ();

Step3:- Use select class methods to select options.

ex:- methods in select clan:-

A) selection methods:-

(1)Select By visibleText()

(2) select By value ()

(3) select By index ()

Example program:-

```java
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class dropdown {

        public static void main(String[] args) throws Throwable {
            WebDriver driver=new ChromeDriver();
            driver.get("https://www.facebook.com");
        driver.findElement(By.xpath("(//a[@role='button'])[2]")).click();
        WebElement day = driver.findElement(By.xpath("//select[@id='day']"));
        Select s=new Select(day);
        s.selectByVisibleText("24");
        Thread.sleep(4000);
        WebElement month = driver.findElement(By.xpath("//select[@id='month']"));
        Select s1=new Select(month);
        s1.selectByVisibleText("Oct");
        WebElement year = driver.findElement(By.xpath("//select[@id='year']"));
        Select s2=new Select(year);
        s2.selectByVisibleText("1999");
        driver.findElement(By.xpath("(//input[@type='radio'])[2]")).click();
        Thread.sleep(3000);
        driver.findElement(By.xpath("_6j mvm _6wk _6wl _58mi _3ma _6o _6v")).click();

        }
```

Keys class :-

Keyboard stroke handling:

Keyboard stock can be handled in two ways.

1) With the help of keys class.

2) With the help of robot class.

Keys class:

- It is an inbuilt class present in org.openqa.seleniumpackage.
- Keys class we have to declare inside sendKeys methods.
- Outside sendKeys methods keys class will not work
- Keys class cannot perform its operations with special charecters .
- Keys class don't have any access on desktop windows.
  - it will reduse the Length of Script.

Syntax:

driver.findElement(By.xpath("//input[@id='
sername']")).sendKeys("admin", Keys. TAB, "manager 1,Keys.ENTER);

Syntax and program :- it is always used in SendKeys ();

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

public class keys {
public static void main(String[] args) {
    WebDriver driver=new ChromeDriver();
    driver.get("https://www.saucedemo.com/");
driver.findElement(By.id("user-name")).sendKeys("standard_user",Keys.TAB,"secret_sauce",Keys.ENTER);
```

Robot Class :-

• To use robot class, we have to create an object for robot class.

• This class is present in java.awt.package In these class two methods are present.

1. Keypress ()

2. Keyrelease ()

• Roboticlass will work with alphabets, no need to use sendKeys method.

It can access in desktop window.

➔

.

Ex program:=

```
import java.awt.Robot;
import java.awt.event.KeyEvent;

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

public class robot {
    public static void main(String[] args) throws Throwable {

        WebDriver driver=new EdgeDriver();
        driver.get("https://www.amazon.com");

        Robot r=new Robot();
        r.keyPress(KeyEvent.VK_PAGE_DOWN);
        r.keyRelease(KeyEvent.VK_PAGE_DOWN);
        Thread.sleep(2000);

        r.keyPress(KeyEvent.VK_PAGE_UP);
        r.keyRelease(KeyEvent.VK_PAGE_UP);

    }

}
```

Differences B/W Keys and Robot :-

S.NO	Keys	Robot
1	It is present is selenium package.	It is present in java.awt.package.
2	Keys class we have to declare inside sendKeys.	Robot class no need to declare inside sendkeys() but we have to create an object of robot class
3	Keys class will not work with special characters	Robot class will work with alphabets.
4	Keys class don't have any access in desktop windows.	Robot class can access in desktop.

Actions Class :-

It is used to perform mouse related actions for elements.

Mouse Action Handling: -

- to perform mouse action in a webpage we have to use action class.
- It is an inbuild class present in selenium package.
- We have to create an object of this class in our script.
- Syntax :- Actions act=**new** Actions();
- To perform mouse action different types of methods are present in action class.

1. moveToElement(target webElement):-

→With the help of this method we can move mouse cursor to the particular webElement.

Syntax:

Act.moveToElement(element);

2. contextClick(right click): -

→To perform right click operation an a particular webElement we have to use this method.

Syntax:=

Act.contextclick ();

3.Build(): -

→With the help of this method we can build a relationship in b/w mouse actions.

Syntax :- act.MoveToElement ().build();

4. dobleClick(): -

→ To perform dobleClick operation in a webpage we have to use this method.

Syntax :- act.dobleClick ();

5. Drag and Drop(source, target):-

→with the help of this method we can drag a element and drop it to the target.

Syntax:-

Act.dragAnddrop(source,target);

6. Perform: -

→With the help of this method we can send the command to the mouse action to perform the operation.

Syntax:-

act.moveToElement(drop).perform();

act.contextClick().perform();

act.doubleClick().perform();

act.dragAndDrop(drag, drop).perform();

Ex program :-

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;

public class AC {
public static void main(String[] args) {
    WebDriver driver=new ChromeDriver();
    driver.get("https://www.google.com");
    WebElement gmail = driver.findElement(By.xpath("(//a[@class='gb_I'])[1]"));
    Actions act=new Actions(driver);
act.moveToElement(gmail).perform();
act.contextClick().perform();
act.doubleClick().perform();
driver.get("https://the-internet.herokuapp.com/drag_and_drop");
WebElement drag = driver.findElement(By.id("column-a"));
WebElement drop = driver.findElement(By.id("column-b"));
act.dragAndDrop(drag, drop).perform();
```

Alerts in Selenium :

(1) void dismiss () :-This method is used when the cancel button is clicked in the alert box

Syntax :- driver. SwitchTo().alert().dismiss();

(2) void accept ():-this method is used to click on the 'ok' button of the alert.

Syntax :- driver. SwitchTo(). alert().accept();

(3) string getText():- This method is used to capture the alert Message.

Syntax:-driver.switchTo (). alert().getext ();

(4) void Sendkeys (String string to send): -this method is used to send data to the alert box.

Syntax:-driver SwitchTo().alert(). Send keys (text);

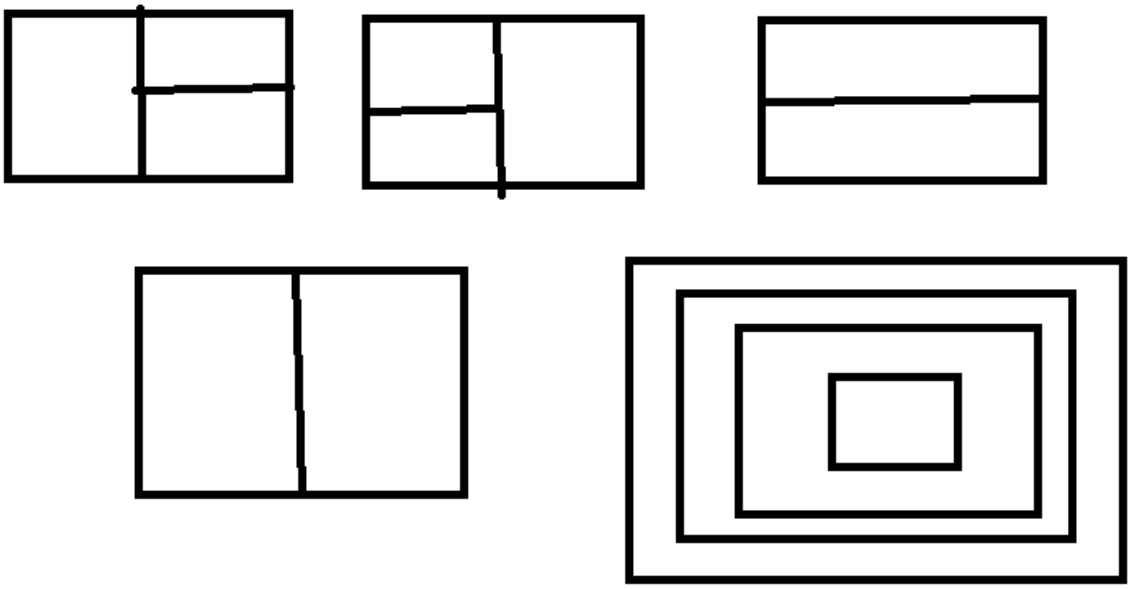
Example program :--

```
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.edge.EdgeDriver;

public class alertPOPUP{
public static void main(String[] args) {
    WebDriver driver=new EdgeDriver();
    driver.get("https://demo.guru99.com/test/delete_customer.php");

    driver.findElement(By.xpath("//input[@name='cusid']")).sendKeys("123456");

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

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

    String text = alt.getText();
    System.out.println(text);

    alt.accept();

    String text2 = alt.getText();
    System.out.println(text2);
```

Iframes :--

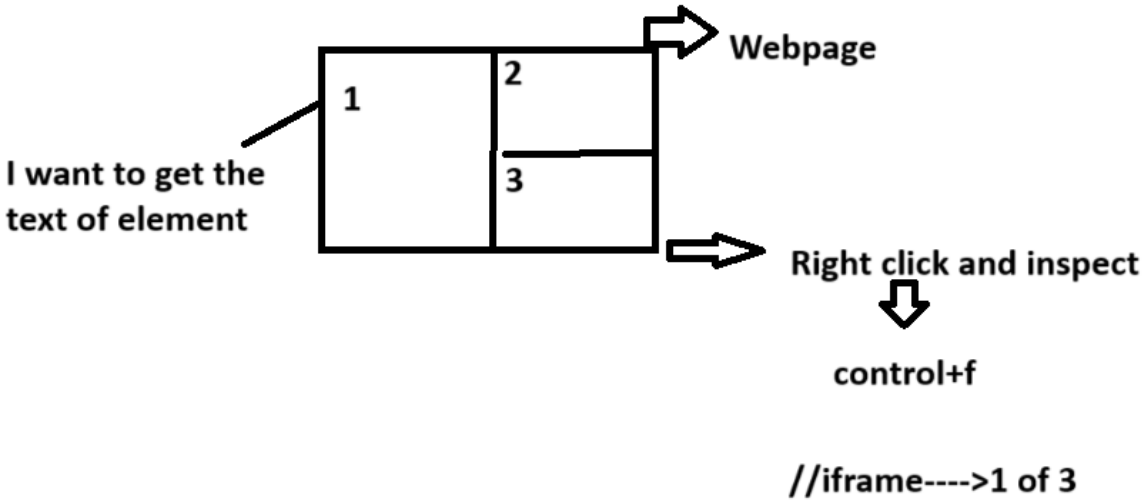
One webpage can be represented by as multiplewebpages.

Structures of frames :-



How to Handle the frames :--

- first of all we should have to switch the control of selenium from main webpage to particular frame.



Q) how to identify the web page is devided into frame ?

(A) //iframe

Q) how to handle the frames ?

(A) driver.SwitchTo ().frame (“ “); → id/name

Instead of id/name if we use any other locator then we will get

“ NoSuchFrameException”.

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.edge.EdgeDriver;


public class iframe {

    public static void main(String[] args) throws Exception {
        WebDriver driver=new EdgeDriver();
        driver.manage().window().maximize();

 driver.get("https://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_submit");
        driver.switchTo().frame("iframeResult");
        driver.findElement(By.xpath("//input[@id='fname']")).clear();

        driver.findElement(By.xpath("//input[@id='fname']")).sendKeys("pavank");

    //driver.switchTo().parentFrame();
    Thread.sleep(3000);
    //driver.switchTo().frame("iframeResult");
    driver.findElement(By.xpath("//input[@name='lname']")).clear();

    driver.findElement(By.xpath("//input[@name='lname']")).sendKeys("mogili");
    //driver.switchTo().parentFrame();
    Thread.sleep(5000);
    //driver.switchTo().frame("iframeResult");
    driver.findElement(By.xpath("//input[@value='Submit']")).click();
}
```

ScreenShot :-

How to take the screenshot in selenim ?

it is phototype with out any functionality first of all we have. to typecast takeScreenshot (i) inside this method getscreenshotAs(). Method is then to Take screenshot.

it takes argument (outputType. FiLE).

and we have to store this screenshot infile. and we have to store in Desktop. Then we have to call the method.

File. Copy(Src, dest);

Ex program:--

```
import java.io.File;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

import com.google.common.io.Files;

public class Screenshot {
public static void main(String[] args) throws Exception {
    WebDriver driver=new ChromeDriver();
    driver.get("https://www.google.com");
    TakesScreenshot ts=(TakesScreenshot) driver;
    File src=ts.getScreenshotAs(OutputType.FILE);
    File dest=new File("C:\\Users\\pavan\\eclipse-workspace\\world\\RestAssured\\screenshot\\photo.png");
    Files.copy(src, dest);
}
}
```

How to take screenshot individual element ?

```java
import java.io.File;
import java.io.IOException;

import org.openqa.selenium.By;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

import com.google.common.io.Files;

public class ScreenshotindividualElement {
    public static void main(String[] args) throws Throwable {
    WebDriver driver=new ChromeDriver();
    driver.get("https://www.google.com");
    WebElement logo = driver.findElement(By.xpath("//img[@class='lnXdpd']"));
    File src = logo.getScreenshotAs(OutputType.FILE);
    File dest=new File("C:\\Users\\pavan\\Desktop\\pavan\\photo1.png");
    Files.copy(src, dest);
```

Limitations Screenshot method :

We can only take screenshot in PNG(Portable Network Graphics) format.

We can't take screenshot of the popup.

We can't take screenshot of multiple browser.

We can't take screenshot of required area on the Web page.

DDF ---Data Driven FrameWork :--

(1) Notepad data fetch :-

Step (1):--Write the data in Notepad

Note Pad :--

```
Username pavan
Password 783104
Email pavank@1222gmail.com
```

write code in selenium with help of third party Tool

Tools are :→

Apache :--

```
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>5.2.2</version>
</dependency>
```

Apache poi-ooxml :-

```
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>5.2.2</version>
</dependency>
```

Ex Program :--

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

public class datareading {

    public static void main(String[] args) throws FileNotFoundException, IOException {

   File f=new File("C:\\Users\\pavan\\Documents\\Username pavan.txt");
        FileInputStream fis=new FileInputStream(f);

        Properties prop=new Properties();
        prop.load(fis);

        System.out.println(prop.getProperty("Username"));
        System.out.println(prop.getProperty("Password"));
        System.out.println(prop.getProperty("Email"));
```

(2) Excel Sheet Data fetch :--

Write data in excel sheet :--

A	B
admin	pavan

Ex program :--

```java
import java.io.FileInputStream;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class excelsheet {



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

        FileInputStream fise=new FileInputStream("C:\\Users\\pavan\\Documents\\pbs.xlsx");
        Workbook wb= WorkbookFactory.create(fise);
        String data=wb.getSheet("Sheet1").getRow(0).getCell(0).getStringCellValue();
        System.out.println(data);
```

Synchronization: -

• Synchronization is nothing but time mismatch in between automation tool speed and web browser speed.

• Synchronization happen because of automation tool speed is faster than web browser speed.

We can handle synchronization issue in 4 ways.

1. Thread.sleep();

2. Implictily wait.

3. Explicit wait.

4. Fluent wait.

1.Thread.sleep();

• It is one of the way to handle the synchronization issue.

• But generally we don't use thread.sleep

• Unnecessary code length will be increased.

- It is not 100% reliable.

Note: -To solve this issue we will go with implicitly wait.

2) Implicit wait: -

• It is one of the way to handle synchronization.

• Implicit wait works very smartly in our text script.

• We have to declare implicit wait only one time just after launching the browser.

Note: -

Implicit wait we can declare more than one time but it is not a recommended approach.

Syntax:-

```
Driver.manage().timeout().implicitywait(Duration.ofseconds(10));
```

• In implicit wait we are giving maximum time interval it means

if any WebElement get loaded before the maximum time,

implicit wait will not wait for the remaining time. It will

perform next operation.

• Implicit wait will work only with findElement and findElements method.

Note:-

During execution if script failed and we are getting exception, we will check with Thread.sleep, that really synchronization issue is present or not, if it is present we will remove thread.sleep and we will go with explicit wait.

Explicit wait: -

- It is one of the way to handle synchronization.

- Explicit wait we can be use more than one time in our script.
- Explicit wait will work with all the methods.
- To use explicit wait we need to create an object of WebDriverWait class.

Syntax:

WebDriver Wait wait=new WebDriver Wait(driver, Duration.ofseconds(10)); Wait until(ExpectedCondition.particularExpectedConditin);

Fluent: -

- Fluent wait is same as explicit wait.
- In case of fluent wait we can customize the time interval.

Difference B/W Implicit wait and Explicit wait :-

S.no	Implicit wait	Explicit wait
1	We have to declare implicit wait only for one time.	We can declare explicit wait multiple time in our script.
2	Implicit wait will work only with findElement and findElements.	Explicit wait will work with all the methods.
3	In case of implicit wait no need to create an object.	In case of explicit wait we have to create webdriverwait class object.
4	Syntax: **Driver manage() timeouts(). Implicitywait(During.ofseconds(10);**	**Syntax:** **Wait.until (ExpectedCondition.particular ExpectedConditin);**

TESTNG

Author :-- Cedric Beust

Writtenin java .

This type Unit Testing Tool.

INTRODUCTION:

➤ It is an open source automated testing framework; where NG of TestNGmeans Next Generation.

➤ TestNG is similar to JUnit but it is much more powerful than JUnit but still it's inspired by JUnit.

➤ It is designed to be better than JUnit, especially when testing integrated classes.

ADVANTAGES OF TESTNG:

➤ It gives the ability to produce HTML Reports of execution.

➤ Annotations made testers life easy.

➤ Test cases can be Grouped & Prioritized more easily.

➤ Parallel execution is possible.

➤ Generates Logs.

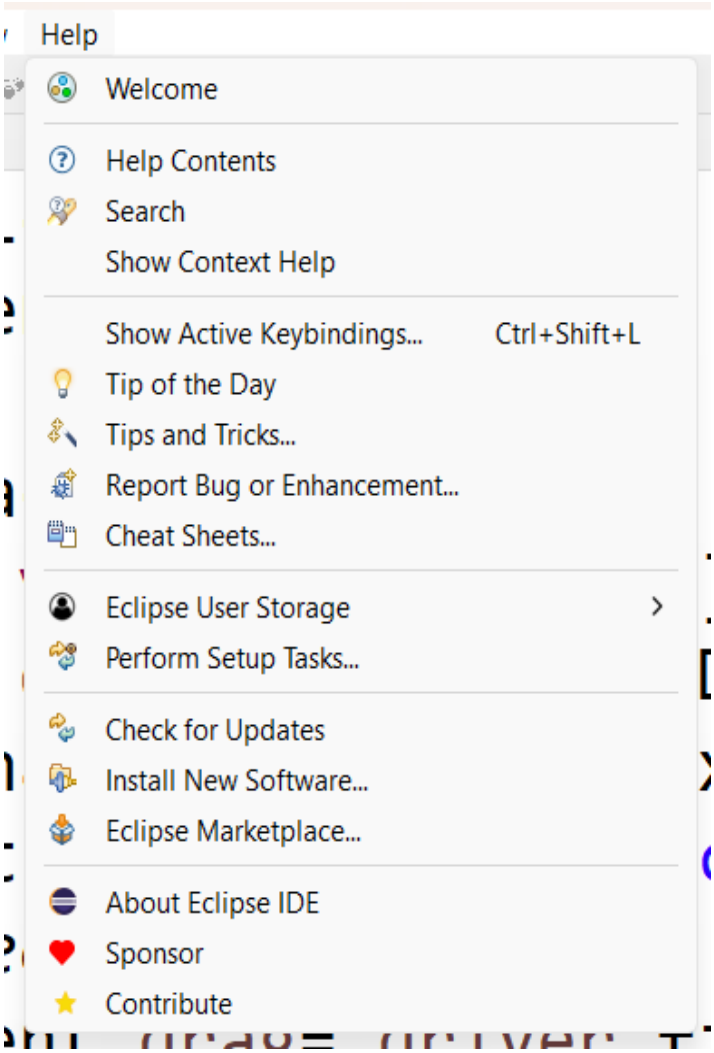
➤ Data Parameterization is possible.

➤ Automatically return the failure test case.

How To install TestNg :-

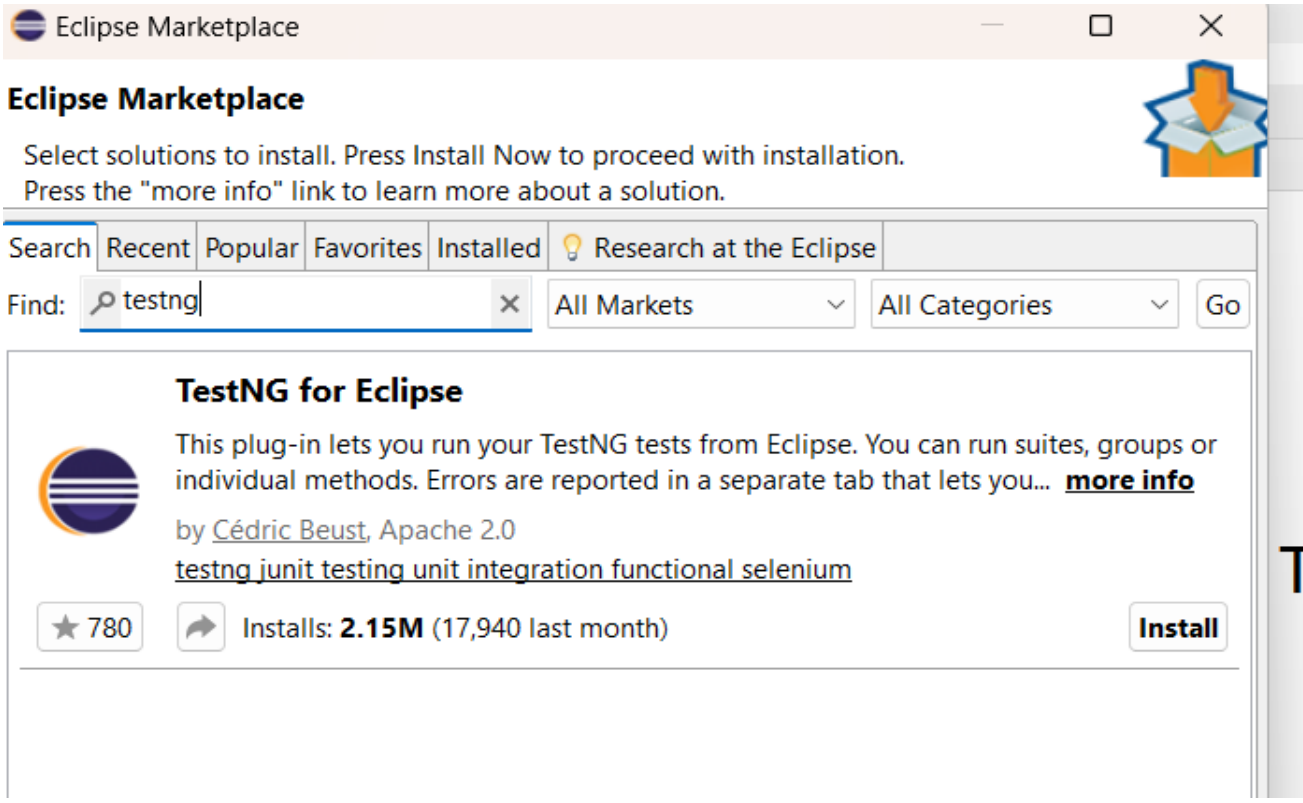
Steps :-

- Creat Maven Project.
- In pom.xml Add some necessary dependencies are version
- Version are 7.10.2
- Install testing in eclipse market.
- Go to help in eclipse



Go to eclipse marketplace

Then search Testng ofter install testNg



How ganarete Reportes in Testing:-

```
package practice;

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

Run All
public class test {
@Test
Run | Debug
public void TC () {
    Reporter.log("boss i am emailablereport",true);
}
}
```

Procedure to creat Report

- Refresh The project (Selenium) → test out put folder Can be ganarated under the project.
- click on Test out put folder we will be having Test Emailable Report. XML
- Right click on emailable Report → open with webbrowser Report you will get

Example Report like this:-

File C:/Users/pavan/OneDrive/Documents/TestNg_pavan/test-output/emailable-report.html

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	1	0	0	0	81		

Class	Method	Start	Time (ms)
Default suite			
Default test — passed			
practice.test	TC	1730703124834	17

Default test

practice.test#TC

Messages
boss i am emailablereport

Annotations in TestNG:

@BeforeSuite: The annotated method will be run before all tests in this suite have run.

@AfterSuite: The annotated method will be run after all tests in this suite have run.

@BeforeTest: The annotated method will be run before any test method belonging to the classes inside the tag is run.

@AfterTest: The annotated method will be run after all the test methods belonging to the classes inside the tag have run.

@BeforeClass: The annotated method will be run before the first test method in the current class is invoked.

@AfterClass: The annotated method will be run after all the test methods in the current class have been run.

@BeforeMethod: The annotated method will be run before each test method.

@AfterMethod: The annotated method will be run after each test method.

@Test: The annotated method is a part of a test case.

Example program in Annotaions :_

File Edit Source Refactor Source Navigate Search Project Run Window Help

Package Explorer

- pavan_automation
- TestNg_pavan
  - src/main/java
  - src/main/resources
  - src/test/java
    - practice
      - sample.java
      - test.java
  - src/test/resources
  - JRE System Library [Java
  - Maven Dependencies
  - src
  - target
  - test-output
    - Default suite
    - junitreports
    - bullet_point.png
    - collapseall.gif
    - emailable-report.htm
    - failed.png
    - index.html
    - jquery-3.6.0.min.js
    - navigator-bullet.png
    - passed.png
    - skipped.png
    - testng-reports.css
    - testng-reports.js
    - testng-reports1.css
    - testng-reports2.js
    - testng-results.xml
  - pom.xml

actclass.java | test.java | sample.java

```
Run All
public class sample {
@BeforeSuite
public void sample() {
 System.out.println("before suite");
}
@AfterSuite
public void sample5() {
System.out.println("after suite");
}
@AfterClass
public void sample6() {
    System.out.println("after class");
     }
 @AfterMethod
 public void sample7() {
 System.out.println("After method");

}
 @AfterTest
public void sample8() {
 System.out.println("after test");

}
 @BeforeClass
 public void sample1() {
 System.out.println("before class");
 }
 @BeforeMethod
 public void sample2() {
 System.out.println("before method");
 }

 @BeforeTest
public void sample3() {
System.out.println("before test");
}
 @Test
Run | Debug
public void sampl4e() {
     System.out.println(" test");
}
```

Out put :-

```
Problems  @ Javadoc  Declaration  Console ×  Results of running cl
<terminated> sample [TestNG] C:\Program Files\Java\jdk-22\bin\javaw.exe  (4 Nc
[RemoteTestNG] detected TestNG version 7.10.2
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementa
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders f
before suite
before test
before class
before method
 test
After method
after class
after test
PASSED: practice.sample.sampl4e

===============================================
    Default test
    Tests run: 1, Failures: 0, Skips: 0
===============================================

after suite

===============================================
Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
===============================================
```

PRIORITY

- We can pass priority to the particular test case.
- We can pass both positive and negative value.
- It will execute based on ascending order.

➤ If we give Same priority then it will execute based on the alphabetic order.

Example program:-

```
package practice;

import org.testng.Reporter;

Run All
public class priority {
@Test
Run | Debug
public void logout() {
    Reporter.log("logout from application",true);
}
@Test(priority=1)
Run | Debug
public void login() {
    Reporter.log("login to application",true);
}
}
```

Out put

```
[RemoteTestNG] detected TestNG version 7.9.0
SLF4J: Failed to load class "org.slf4j.impl.StaticLogg
SLF4J: Defaulting to no-operation (NOP) logger impleme
SLF4J: See http://www.slf4j.org/codes.html#StaticLogge
Logout from application
Login to application
PASSED: practice.priority.logout
PASSED: practice.priority.login

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


===============================================
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
===============================================
```

INVOCATION COUNT:

➤ If you want to run the particular test case to run for many times, We can use one method called invocation test case.

➤ It will run the test case for that particular times.

Program:-

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

Run All
public class invocationcount {
@Test (invocationCount=10)

Run | Debug
public void demo() {
    Reporter.log("hi good mrg",true);
}
}
```

Out put :--

```
SLF4J: See http://www.slf4j.org/codes.html#StaticLo
hi good mrg
hi good mrg
hi good mrg
hi good mrg
hi good mrg
hi good mrg
hi good mrg
hi good mrg
hi good mrg
hi good mrg
PASSED: practice.invocationcount.demo
PASSED: practice.invocationcount.demo
PASSED: practice.invocationcount.demo
PASSED: practice.invocationcount.demo
PASSED: practice.invocationcount.demo
PASSED: practice.invocationcount.demo
PASSED: practice.invocationcount.demo
PASSED: practice.invocationcount.demo
PASSED: practice.invocationcount.demo
PASSED: practice.invocationcount.demo

===============================================
    Default test
    Tests run: 1, Failures: 0, Skips: 0
===============================================
```

IGNORING THE TEST CASE:

➤ For ignoring the test case We can use one method called Enabled.

➤ When we use enabled=false, It will skip the particular test case.

Program:-

```java
import org.testng.Reporter;

Run All
public class enabled {
@Test
Run | Debug
public void m1() {
    Reporter.log("hi good mornig",true);
}
@Test(enabled = false)
Run | Debug
public void m2() {
    Reporter.log("hi good ",true);
}
@Test
Run | Debug
public void m3() {
    Reporter.log("hi  mornig",true);
}
@Test
Run | Debug
public void m4() {
    Reporter.log(" good mornig",true);
}
}
```

Out put:-

```
hi good mornig
hi  mornig
 good mornig
PASSED: practice.enabled.m1
PASSED: practice.enabled.m3
PASSED: practice.enabled.m4

===============================================
    Default test
    Tests run: 3, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 3, Passes: 3, Failures: 0, Skips: 0
===============================================
```

Time Outs :-

- Whenever one of test case is taking to much time

it is used to give the particular time Exicution.

Program :-

```
import org.testng.Reporter;

Run All
public class timeouts {
@Test
Run | Debug
public void m1() {
    Reporter.log("hello",true);
}
@Test (timeOut=10)
Run | Debug
public void m2() {
    Reporter.log("hii",true);
}
@Test
Run | Debug
public void m13() {
    Reporter.log("by",true);
}
@Test
Run | Debug
public void m4() {
    Reporter.log("hello",true);
}
```

Out put :-

```
hello
by
hii
hello
PASSED: practice.timeouts.m4
PASSED: practice.timeouts.m2
PASSED: practice.timeouts.m1
PASSED: practice.timeouts.m13

===============================================
    Default test
    Tests run: 4, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 4, Passes: 4, Failures: 0, Skips: 0
===============================================
```

DependsOnMethod :-

if login fails then logout also failes because they are dependents.

Program :-

```java
import static org.testng.Assert.assertEquals;

Run All
public class dependsonmethod{
@Test(dependsOnMethods = "login")
Run | Debug
public void logout() {
    Reporter.log("logout from application",true);
}
@Test
Run | Debug
public void login() {
    Reporter.log("login to application",true);

}
}
```

OutPut:-

```
login to application
logout from application
PASSED: practice.dependsonmethod.login
PASSED: practice.dependsonmethod.logout

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


===============================================
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
===============================================
```

paralled execution:-

it is process of running multiple testcase Parallely Eather than one after one.

Here we can open two Applications Simultaneously Same Time on same browser at same Time.

a.java

```java
package parallel;

import org.openqa.selenium.WebDriver;

Run All
public class a {
      WebDriver driver=new ChromeDriver();
       @Test
       Run | Debug
       public void tc1() {
       driver.get("https://www.facebook.com");
       }
}
```

b.java

```java
package parallel;

import org.openqa.selenium.WebDriver;

Run All
public class b {
      WebDriver driver=new ChromeDriver();
      @Test
      Run | Debug
      public void tc2() {
      driver.get("https://www.instagram.com");
      }
}
```

Ofter write script select two clases (control hold it)

If select the clases

Right click ->go to TestNg ->converting to TestNg -> ofter ctreat testing.xml suite change to suite name parrel.xml.

Liki this →

```
https://testng.org/testng-1.0.dtd (doctype)
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
<test thread-count="5" name="Test1">
<classes>
<class name="pavan_klayandemo.a"/>
 </classes>
 </test> <!-- Test -->
<test thread-count="5" name="Test2">
<classes>
<class name="pavan_klayandemo.b"/>
</classes>
</test> <!-- Test -->
</suite> <!-- Suite -->
```

Batch Execution :-

it is process of running multiple classes with Single click is called as batch Execution.

sample1.java

```
package batchExecutiom;

import static org.testng.Assert.fail;

Run All
public class sample1 {
@Test
Run | Debug
public void Tc1() {
    Reporter.log("i am Tc1",true);
    Assert.fail();
}
@Test
Run | Debug
public void Tc2 () {
    Reporter.log("i am Tc2",true);
}
@Test
Run | Debug
public void Tc3 () {
    Reporter.log("i am Tc3",true);
}
}
```

sample2.java

```
package batchExecutiom;

import org.testng.Reporter;

Run All
public class sample2 {
    @Test
    Run | Debug
    public void Tc4() {
        Reporter.log("i am Tc4",true);
    }
    @Test
    Run | Debug
    public void Tc5 () {
        Reporter.log("i am Tc5",true);
    }
    @Test
    Run | Debug
    public void Tc6 () {
        Reporter.log("i am Tc6",true);
    }

}
```

sample3.java

```
package batchExecutiom;

import org.testng.Reporter;

Run All
public class sample3 {
    @Test
    Run | Debug
    public void Tc7() {
        Reporter.log("i am Tc7",true);
    }
    @Test
    Run | Debug
    public void Tc8 () {
        Reporter.log("i am Tc8",true);
    }
    @Test
    Run | Debug
    public void Tc9 () {
        Reporter.log("i am Tc9",true);
    }
    }
```

Procedure for batch execution :-

Select All clases {Control hold it }if select all clases .

Right click on all clases ->go to TestNG ->Convert to TestNG

Ofter creat suite testing.xml change the name batch.xml

```
https://testng.org/testng-1.0.dtd (doctype)
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Test">
    <classes>
      <class name="practice.A"/>
      <class name="practice.B"/>
      <class name="practice.C"/>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

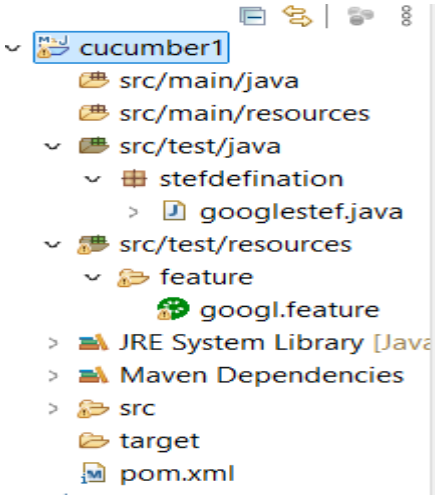
Out put :-

```
<terminated> TestNg_pavan_batch.xml [TestNG] C:\Program Files\Java\jdk-22\bin\jav
[RemoteTestNG] detected TestNG version 7.10.2
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementatic
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for
i am Tc1
i am Tc2
i am Tc3
i am Tc4
i am Tc5
i am Tc6
i am Tc7
i am Tc8
i am Tc9

===============================================
Suite
Total tests run: 9, Passes: 9, Failures: 0, Skips: 0
===============================================
```

Cucumber Frame Work :-

Steps for Cucumber frame Work :-

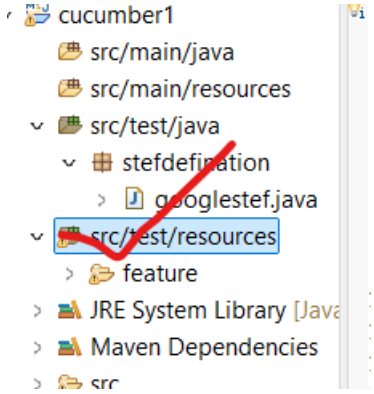
step 1: Create a new Maven project



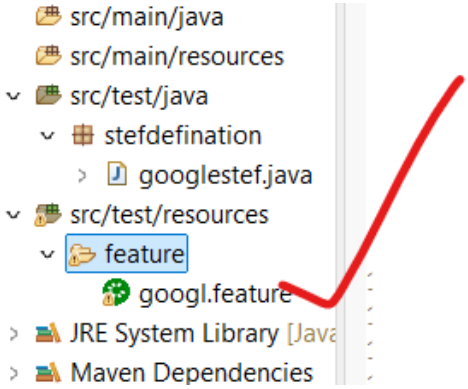
step 2:- Add Maven dependencies Cucumber Java (Version – 7.20.1)/Cucumber Junit (Version—7.20.1)/ Junit/selenium java.

```
<project xmlns="https://maven.apache.org/POM/4.0.0" xmlns:xsi="https://www.w3.org/200
  <modelVersion>4.0.0</modelVersion>
  <groupId>cucumber1</groupId>
  <artifactId>cucumber1</artifactId>
  <version>0.0.1-SNAPSHOT</version>

   <dependencies>

  <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.26.0</version>
</dependency>

<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
<dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-java</artifactId>
    <version>7.20.1</version>
</dependency>
<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-junit -->
<dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-junit</artifactId>
    <version>7.20.1</version>
    <scope>test</scope>
</dependency>

  </dependencies>
</project>
```

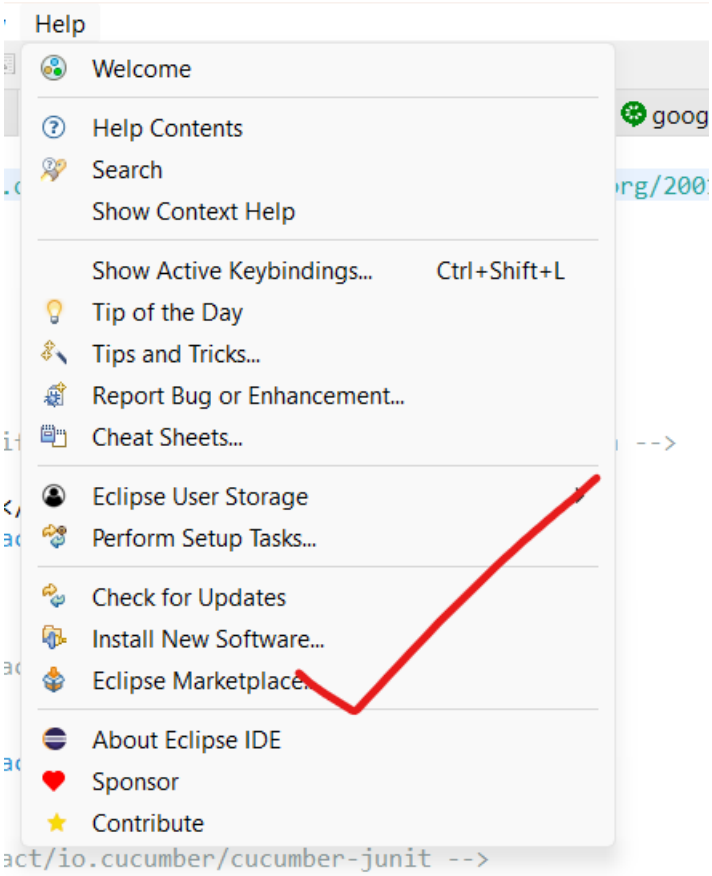
step 3 :- Create a folder Features under Src/Test/Resource.



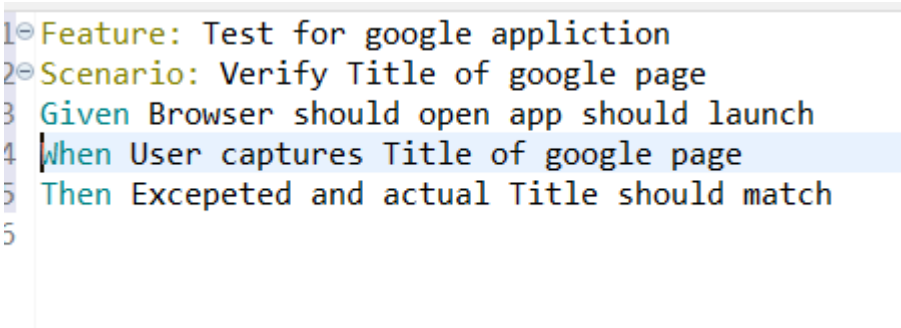
Step 4: Under features folder Create a new feature file login feature.



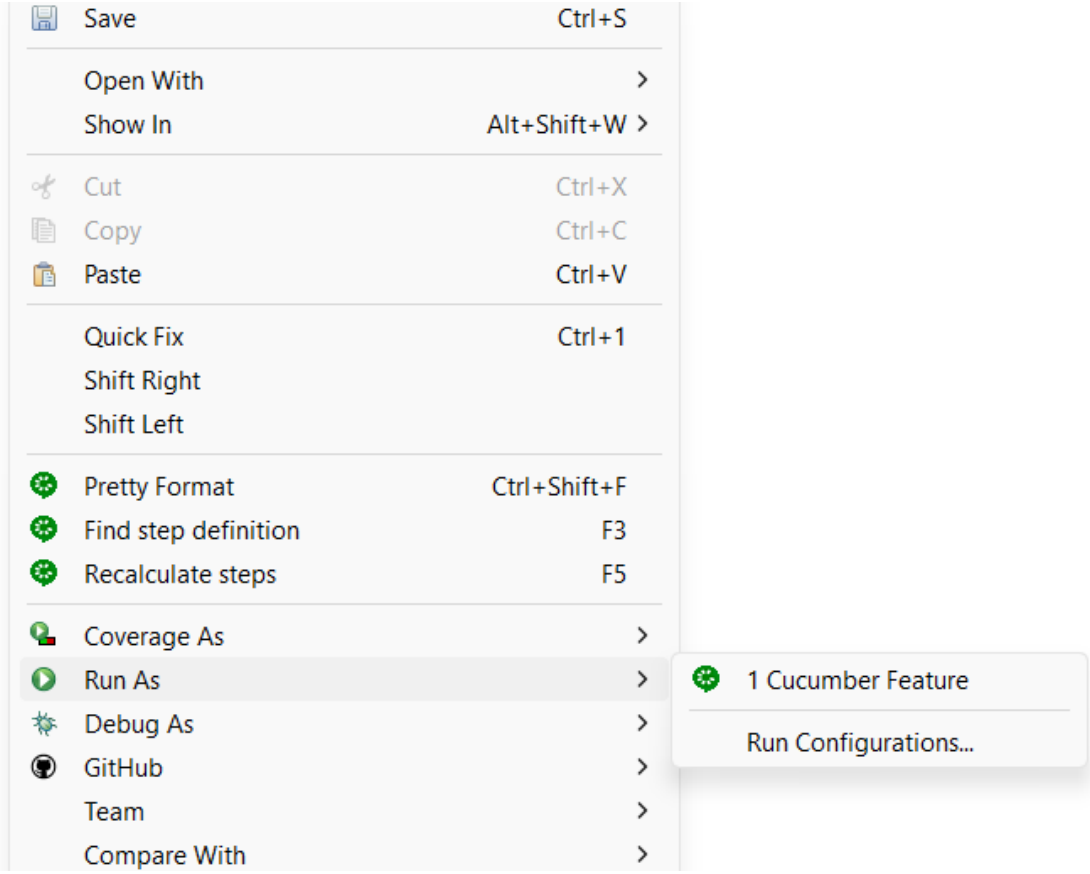
Step 5:- Download Cucumber plugin from Marketplace.



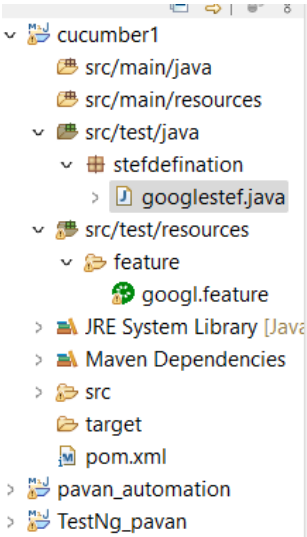
Step 6:- create feature file and contents.



Step 7: Try to Run The feature file.



Step 8: Add step Definitions (Glue Code under src/ test/Java Package)




```java
package stefdefination;

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

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class googlestef {
    WebDriver driver;
    public String actT;
    @Given("Browser should open app should be launch")
    public void browser_should_open_app_should_be_launch() {
       driver=new ChromeDriver();
       driver.get("https://www.google.com");
    }

    @When("User Capture Title of google page")
    public void user_capture_title_of_google_page() {
       actT=driver.getTitle();
    }

    @Then("Expected and Actual Title Sharld Match")
    public void expected_and_actual_title_sharld_match() {
        if (actT.contains("Google")) {
            System.out.println("tc is pass");

        }
        else {
            System.out.println("tc is fail");
        }
    }
}
```

Step 9 : Create a Runner class.

```java
package runnerclass;


    import org.junit.runner.RunWith;

    @RunWith(Cucumber.class)
    @CucumberOptions(features="C:\\Users\\pavan\\OneDrive\\Documents\\cucumber1\\src\\test\\resources\\feature",glue="stefdefination")
    public class google_runner {

    }
```

Output :-

Tc is pass