Selenium

There was a company called ThoughtWorks, who invented Selenium tool.

Initially they have invented Selenium Core & it was not commercially released to the market.

The following selenium tools which were released to the market:

- **Selenium IDE** (Integrated Development Environment)

    1. It is a plugin/add-on for firefox browser only.
    2. It is supporting record and play back.
    3. It understands selenese language

- **Selenium RC** (Remote Control)

    1. It supports all browsers
    2. It was a server which needs to be started to work with RC.
    3. No record and play back.

- **Selenium GRID:** It is used to run the selenium scripts parallel across different OS OR PC's.

This tool is used to run the selenium scripts either in the same computer or in the remote computer. This supports parallel execution across different platforms

By this time Google in collaboration with ThoughtWorks => released a new Selenium version viz., - Selenium WebDriver

Selenium WebDriver (Functional Testing tool) tools understands following programming languages (language bindings):

-Core Java, -C#-Python-Ruby-JavaScript

Q: Why automations?

Ans: 1. Reusability

2. Consistency in test execution

3. Better test coverage

4. Reduces testing duration over period of time

5. Reduces cost over a period of time.

6. Reduces head count over a period of time.

7. Over night execution (Nightly Execution)

8. Helps to caught more defects

Q: Why selenium webdriver?

Ans:

1. Its free & open source

2. Supports multiple browsers

3. Supports multiple OS (windows, Linux, Mac)

4. Supports multiple programming languages (java, C#, Ruby, Python &javaScript)

5. Parallel (Distrubuted testing) execution can be achieved across different browsers & OS

6. As it supports programming languages we can achieve lot of things by writing customized codes. Ex: Image comparision.

Q: Drawbacks of selenium webdriver?

1. As its free & open source we don't get ready vendor support (But helping comminities are there)

2. As it supports Programming languages we need highly skilled resources

3. There is no reporting mechanism (we can use third party tools with selenium)

4. It doesnot supports image OR bitmap comparision

5. It supports only web applications. It doesnot supports window OR desktop application (we can use third party tools)

6. It doesnot identify the browsers which are opened manually.

7. It doesnot supports Unix OS

Q: Selenium Architecture?

Ans: It has a highly scalable architecture.

Selenium Webdirver is a interface (api).

Selenium webdriver is a interface. It has many child class which have implemented all the abstract methods available in the interface using different language bindings.

When use any language bindings (Ex: Java) it has to interact with browser driver classes. In order for the language bindings to interact with the driver classes they uses a RestFULL service viz., Jason wire protocol.

Browser Driver classes with the help of Json wire Protocol understands java commands and perform the same on the real browsers using HTTP over HTTP server request.

https://www.selenium.dev

How to set up the selenium webdriver:

1. download the language binding (.zip file), extract, get the jar files & set the build path in eclipse

2. Install the real browsers

3. download Third party drivers (plugins), bindings for browsers

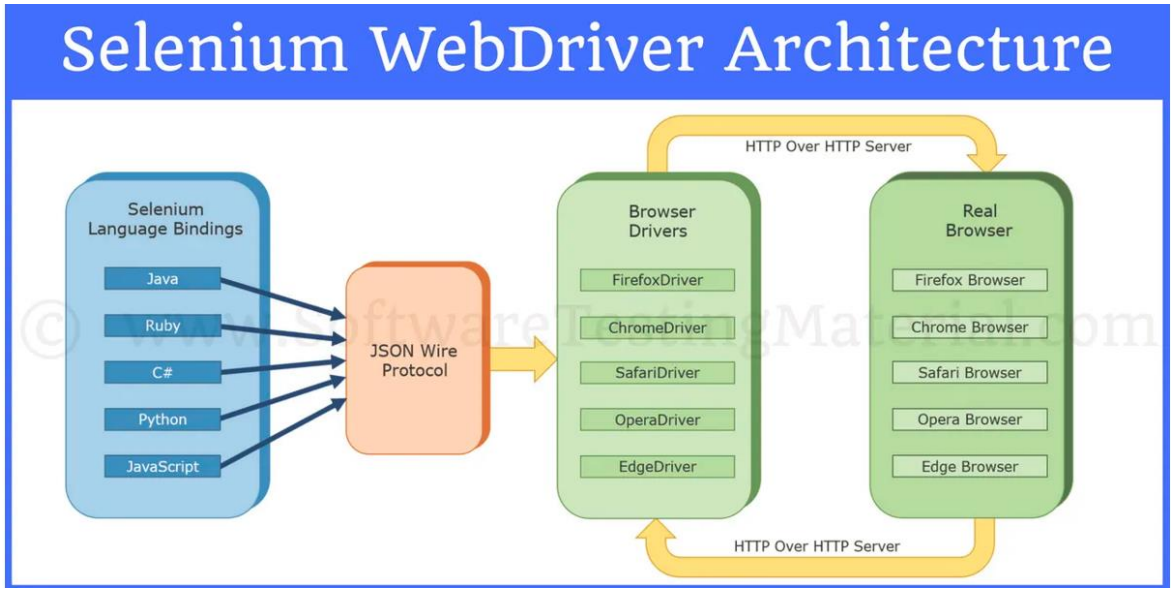
Ex: For chrome browser--> chromedriver.exe file (32 bit)

forfirefox browser--> geckodriver.exe file (32 bit & 64 bit)

for IE browser --> ieDriverServer.exe (32 bit)

for Edge browser --> Edgedriver.exe

4. Need to have one web application to automate it. (ActiTime)



What are the difference between Selenium IDE, RC and Webdriver?

Feature	Selenium IDE	Selenium RC	WebDriver
Browser Compatibility	Selenium IDE comes as a Firefox plugin, thus it supports only Firefox	Selenium RC supports a varied range of versions of Mozilla Firefox, Google Chrome, Internet Explorer and Opera	WebDriver supports a varied range of versions of Mozilla Firefox, Google Chrome, Internet Explorer and Opera. Also supports HtmlUnitDriver which is a GUI less or

			headless browser.
Record and Playback	Yes	No	No
Server to be started before executing the test scripts	No	Yes	No
Architecture	Selenium IDE is a Javascript based framework	Selenium RC is a JavaScript based Framework	WebDriver uses the browser's native compatibility to automation
Object Oriented	Selenium IDE is not an object oriented tool	Selenium RC is semi object oriented tool	WebDriver is a purely object oriented tool
Dynamic Finders (for locating web elements on a webpage)	Selenium IDE doesn't support dynamic finders	Selenium RC doesn't support dynamic finders	WebDriver supports dynamic finders
Handling Alerts, Navigations, Dropdowns	Selenium IDE doesn't explicitly provides aids to handle alerts, navigations, dropdowns	Selenium RC doesn't explicitly provides aids to handle alerts, navigations, dropdowns	WebDriver offers a wide range of utilities and classes that helps in handling alerts, navigations, and dropdowns efficiently and effectively.
WAP (iPhone/Android) Testing	No	No	WebDriver is designed in a way to efficiently support testing of iPhone/Android applications. The tool comes with a large range of drivers for WAP based testing. For example, AndroidDriver, iPhoneDriver
Listener Support	No	No	WebDriver supports the implementation of Listeners

Speed	Selenium IDE is fast as it is plugged in with the web-browser that launches the test. Thus, the IDE and browser communicates directly	Selenium RC is slower than WebDriver as it doesn't communicates directly with the browser; rather it sends selenese commands over to Selenium Core which in turn communicates with the browser.	WebDriver communicates directly with the web browsers. Thus making it much faster

HTML: Hyper Text Markup Language

- It is platform independent
- It is mainly used for presentation purpose
- It is not case sensitive. But Lower case is highly recommended
- It contains Tags. Tags are also k.a., Elements.
- All the tags should be properly nested.
- All the tag names are predefined.
- All the tags should be closed properly
- The tags will contains attribute name & attribute values.
- HTML doesnot preserve white space
- To comment html code we use below syntax:

<!-- html source code -->

Common tags/Elements used &Its purposes:

(1) <html> - Its a root tag

(2) <head> - Used to write javascript& CSS style

(3) <title> - to provide Page title

(4) <body> - contains visible content of the page

(5) <p> - for paragraph(6) <pre> - for Paragraph

(7) <hr> - to generate horizontal line

(8)
 - to generate new line, (9) <b> - to make bold

(10) <input> - to create button, text field, checkbox & radio button

(11) <button> - to create a button, (12) <script> - to write javascript

(13) <style> - for writing css style

(14) - To write text, para& style

(15) - to write text, label etc, (16) <form> - it acts as a container

(17) <a> - for link, (18) <img> - images

(19) <textarea> - for multi edit text field

(20) <iframe><frame> - to insert one page inside another page

(21) <select> - to create a drop down

(22) <option> - to create elements inside the drop down

(23) <table> - to create a webtable

(24) <tr> - table rows, (25) <th> -Table header

(26) <td> - table data/cell, (27) - Ordered list

(28) - Unordered list, (29) - List items

(30) <h1> to <h6>: There are 6 different headers in html

There are 2 types of elements:

1. Inline elements: Inline elements display in a same line.

Ex: <a>, <input>etc

2. Block level elements: Block level elements display in a next line.

Ex: <h1> to <h6>, <p>, <pre>etc

Note: As per w3 (World Wide Web) each tags contains many attributes.

1. How to create paragraphs & Headers?

2. How to create edit/Text fileds, Buttons?

3. How to create check box OR radio buttons?

4. How to create links?

5. How to create dropdown lists?

6. How to create images?

7. How to create webtables?

8. How to create Frames OR iframes

Q: How to create html files?

A:Open notepad-> save as <FileName>.html->Save as Type All Files(*)

9. How to apply Style attributes?

CSS (Cascading Style Sheet):

CSS is used to define the style to the html elements.

Syntax:

style="<attributrName>:<attributeValue>"

Different ways of defining CSS:

(a) Inline Approach: Defining the style within the each tag indicates inline approach.

Drawbacks:

1. No reusability 2. more line of codes

(b) Internal Approach: Defining the styles inside the <head> section of the each html pages by using <style> tag.

Here we use id & class attributes to defferenciate the tags uniquely.

- id . - class

Advantages: Reusability within the same page

Drawbacks: No reusability between the pages

(c) External Approach: defining the style attributes in the external files. The file extension should be .css

This external file will be called inside the html page whereever required. Thus increases the reusability within & as well as between the pages.

9. How to write javascript in the html pages?

Xpath:

XPath is defined as XML path. It is a syntax or language for finding any element on the web page using XML path expression. XPath is used to find the location of any element on a webpage using HTML DOM structure

Types of Xpath:

1) Absolute Xpath, 2) Relative Xpath

1) Absolute Xpath (AX):

- It starts from the root element (Ex: html)

- It starts with "/"

Q: Enter First username using AX?

Ans: /html/body/form/input

2) Relative Xpath (RX)

-It will not start from root element

-It starts with "//"

(A) Using tagName

Q: Enter first user (First Edit) name using RX?

Ans: //input

Q: Enter first password (Second edit) using RX?

Ans: //input[2]

(B) Xpath with Attribute name & values?

syntax:

//<tagName>[@<attrName>=<attrValue>]

//input[@id='un1_id1']

//input[@name='pwd1_name1']

(C) Xpath with more than one Attribute name & values?

Ans:
//<tagName>[@<attrName1>=<attrValue1>][@<attrName2>=<attrValue2>]

//input[@id='ok_btn_id1'][@name='ok_btn_name1']

(D) Xpath with and logical operator?

Ans: //<tagName>[@<attr1>=<val1> and @<attr2>=<val2>]

//input[@id='ok_btn_id1' and @name='ok_btn_name1']

(E) Xpath with or logical operator

//<tagName>[@<attr1>=<val1> or @<attr2>=<val2>]

//input[@id='ok_btn_id1' or @name='xxxxxxxx']

(F) Xpath using text() method?

Ans: //<tagName>[text()=<value>];

//p/b[text()='Sample paragraph Two']

//h1[text()='Header One']

Q: When to use .getText() method to read the text on elements?

Ans: If the text is located on below tags/Elements then use .getText()

<h1> to <h6>, <p>,<pre>, ,,<th>,<td>,<a>,<img>,<label>

Q: When to use .getAttribute("") method to read the text & attributes present on the elements?

Ans: - to read all attributes

- button with <input> tag: .getAttribute("value");

- Text inside the text field: .getAttribute("value");

- Text inside the TextArea: .getAttribute("value");

- Read the image src name: .getAttribute("src");

- Read the id value for UN: .getAttribute("id");

Q: How to read the selected text in the dropdown?

Ans: .getFirstSelectedOption().getText()

Q: Xpath with partial matches for dynamic elements?

It is used when attribute values are changing during runtime (Dynamic element/object).

(a) starts-with(@<attrName>, <attrValue>)

(b) ends-with(@<attrName>, <attrValue>)

(c) contains(@<attrName>, <attrValue>)

(a) starts-with(): It is used when the element atributes last values are changing frequently.

syntax: //<tagName>[starts-with(@<attr>, <starting constant Value>)];

//input[starts-with(@id,'un_text_id')]

(b) ends-with(): It is used when the element attributes initial values are changing frequently.

syntax: //<tagName>[ends-with(@<attr>, <EndingValue>)];

//input[ends-with(@id,'_id1234')]

(c) contains(): It is used when the element attributes initial/middle/end values are changing frequently. It acts as starts-with(), ends-with() & contains().

syntax: //<tagName>[contains(@<attr>, <start/end/middle>)];

//p/b[contains(text(),'One')]

//h1[contains(text(),'One')]

//input[contains(@id,'t_id12')]

Xpath Axis:

(a) following-sibling (b) following (c) preceding-sibling

(d) preceding (e) descendant (f) ancestor (g) parent (h) child

(a) following-sibling: It travels in forward direction within one row OR within next level.

Q: Identify the designation for kalam?

//<tagName>[@<attr>=<val>]/following-sibling::<nextLevel>;

//td[text()='Abdul Kalam']/following-sibling::td[1]

Q: Enter salary for Modi?

//td[text()='Narendra Modi']/following-sibling::td/input

(b) following: It travels in forward direction within one row or many rows OR multiple levels.

The following can acts as following-sibling & following.

Q: Identify the designation of the person who is after Modi?

Ans: //td[contains(text(),'Modi')]/following::tr[1]/td[3]

Q: select the gender "M" who is next to KALAM?

Ans: //td[contains(text(),'Kalam')]/following::tr[1]/td[6]

(c) preceding-sibling: It travels in a backward direction within one row OR one level.

Q: Find the name of the indian Scientist?

Ans: //td[contains(text(),'Scientist')]/preceding-sibling::td[1]

Q: Identify the person name who's gender is Female?

Ans: //td[text()='Female']/preceding-sibling::td[4]

(d) preceding: It can travel in backward direction within one row OR multiple rows OR multiple levels and upward directions.

Q: Find the person name who is before Narendra Modi?

//td[text()='Narendra Modi']/preceding::tr[1]/td[2]

Q: Enter salary to the person who is before Indian Cricketer?

Ans: //td[text()='Indian Cricketer']/preceding::tr[1]/td[5]/input

(e) descendant: It goes in a forward direction to the child element

Q: Enter salary to the person who is present in 2nd row?

Ans: //tr[2]/descendant::td[6]

//table/descendant::tr[2]/descendant::td[6]

(f) ancestor: It travels in backward direction from child element towards parent element

Q: Find the row OR table for the given value?

//td[text()='Rahul Dravid']/ancestor::tr[1]

//td[text()='Rahul Dravid']/..

//td[text()='Rahul Dravid']/ancestor::table

//td[text()='Rahul Dravid']/../..

in Xapth, the tagname & attribute name can be substituted by regular expression characters.

//*[@id='un_text_id1']

//input[@*='un_text_id1']

Q: Xpath using parent hierachy?

Ans: //form[@id='frm2']/input[@id='frm1_un_id1']

CssSelector:

A CSS Selector is a combination of an element selector and a value which identifies the web element within a web page. They are string representations of HTML tags, attributes, Id and Class. As such they are patterns that match against elements in a tree and are one of several technologies that can be uses to select nodes in an XML/html document.

There are 2 types:

(a) Absolute CssSelector (b) Relative CssSelector

(a) Absolute CssSelector:

- It starts from the root element
- It will not start with "/"

Q: Enter first UserName values?

Ans: html body form input

Q: Enter first Password values?

Ans: html body form input[id='frm1_pwd_id1']

Q: Enter second User name value?

Ans: html body form[id='frm2'] input

Q: Enter second Password value?

Ans: html body form[id='frm2'] input[id='frm1_pwd_id1']

(b) Relative CssSelector:

- It will NOT start from the root element
- It will NOT starts with "//"

Q: Enter first username value?

Q: Enter first password value?

input#frm1_pwd_id1

#frm1_pwd_id1

Q: cssSelector with attribute name & value?

<tagName>[<attrName>=<attrVal>]

input[name='frm1_un_name1']

Q: cssSelector with more than one attribute name & value?

<tagName>[<attr1>=<Val1>][<attr2>=<Val2>]

input[id='frm1_un_id1'][name='frm1_un_name1']

Q: cssSelector with Partial Match

(a) ^ (starts-with) (b) $ (ends-with) (c) * (contains)

(a) ^ (starts-with):

Q: Enter first username value?

<tagName>[<attrName>^=<attrValue>]

input[id^='frm1_un_i']

(b) $ (ends-with)

Q: Enter first username value?

<tagName>[<attrName>$=<attrValue>]

input[id$='1_un_id1']

(c) * (contains)

Q: Enter first username value?

<tagName>[<attrName>*=<attrValue>]

input[id*='1_un_id']

nth child concept in cssSelector:

(a) first-child, (b) last-child (c) nth-child(<index>)

(a) first-child:

Q: Find the first child element inside the form one?

form[id='frm1']> :first-child

(b) last-child:

Q: Find the last child element inside the form one?

form[id='frm1']> :last-child

(c) nth-child(<index>):

It counts all the tags in between.

Q: Find the middle child elements inside the form one?

form[id='frm1']> :nth-child(5)

Advantage of cssSelector?

1. It is faster compare to xpath especially in the IE browser

2. In IE cssSelector suits better than the Xpath. Bcoz IE browsers doesnot have well defined xpath engine.

Dis-Advantages:

1. It doesnot supports index.

2. It doesnot support 'and' & 'or' logical operators

3. It doesnot support text() method.

4. It doesnot travels in all the directions unlike xpath

Q: What is alert?

Ans: It is a javascript output statement

Q: What is java output statement?

Ans: System.out.println(); System.out.print();

Q: Types of alerts in javascript?

(a) alert: ex: alert("message");

(b) confirm: Ex: confirm("click button");

it returns boolean value. true if clicked OK button and false if clicked CANCEL

(c) prompt: Ex: prompt("Enter value", "SG Testing");

Q: How to handle the alerts using selenium webdriver?

In selenium webdriver, to handle the alerts we have an interface viz., Alert.

This Alert interface has 4 methods:

(a) .getText() : to read the text present on the alert

(b) .sendKeys() : to enter the values in the prompt

(c) .accept() : to click the +ve buttons viz., Yes, OK, Accept, Confirm etc

(d) .dismiss() : to click the -ve buttons viz., No, Cancel, Reject, Dismiss etc

Note: The browser driver classes will implement the alert interface abstract methods.

Q: How to switch to alert?

```
ChromeDriver ch = new ChromeDriver();

Alert oAlert = ch.switchTo().alert();
```

How to handle the :child winodws in sample applicaion

2. Child windows OR child browsers OR popup browsers

Child windows means upon clicking button/link etc one more browser will open (either in tab/independently) which is k.a., child window/child browser/browser popup.

Whenever selenium opens the browser, it will give a unique number (alpha-numeric) to the browser. which is k.a., window id. Selenium uses this window id to recognize the browser. When child browser opens, that also having window id. So we need to read the child browser window id and switch to child browser using the child browser window id.

Q: How to read the window id's of parent and child browser?

String sParent = oBrowser.getWindowHandle();

Note: getWindowHandle() returns only parent browser window id in the form of String.

Set<String> oChild = oBrowser.getWindowHandles();

Note: getWindowHandles() returns both parent & child browser window ids in the form of Set<String>.

Q: How to switch to the browsers?

oBrowser.switchTo().window(<windowId>);

Q: What is frame OR iFrame?

Ans: In a web technology inserting one page inside the another makes a frame.

Q: How to check the frame is present or not in the web page?

case1: go to dev tool, press Ctrl+F and write the xpath for frame using tagename & check the count. Ex: //iframe

case2: go to the DOM structure and check how many open <html> tags are present. Each open <html> tag indicates a frame. Ex: //html

case3: right click and check for frame text in the context menu. If the context menu contains a word frame, which mean the page contains a frame.

This works only for Chrome & Firefox.

Q: Can we directly perform action on the frames using selenium?

Ans: No. When we create a object to the browser using selenium, By default the control will be on the main browser. Hence we can't perform

action on the frames. So we need to switch to the perticular frame in order to perform action on them.

Q: How to switch to frames using selenium webdriver?

Ans: There are 3 ways we can switch to the frames in selenium webdriver.

(a) <browserObject>.switchTo().frame(<index>);

(b) <browserObject>.switchTo().frame(<name/id>);

(c) <browserObject>.switchTo().frame(<Frame WebElement>);

Q: In independent frame, can we directly move from one frame to another frame in selenium Webdriver?

Ans:No. We can't travel from one frame to another frame directly. First we need to come back to the main page and then switch to another frame.

Q: How to come back to the main page using selenium webdriver?

Ans: <browserObject>.switchTo().defaultContent();

Q: In nested frames, how to switch to parent frame in selenium webdriver?

Ans: <browserObject>.switchTo().parentFrame();

Window components: We can't handle window components using selenium webdriver. Bcoz selenium webdriver does not supports window components.

Hence we have to use thrid party tools viz., AutoIT & Sikuli

AutoIT: It is an independent functional testing tool. It supports only window/desktop/standalone based applications. It doesnot supports web applications.

Q: What is the extension for autoIT scripts?

Ans: the autoIt script extension is .au3

Q: Does autoIt script is case sensitive?

Ans:Yes

Q: How to download & install autoIT?

Ans: i. go to google and search for "Download autoIT"

ii. click on the first official link of autoIT (https://www.autoitscript.com)

iii. do down to 'AutoIt Full Installation' section

iv. download 'AutoIt Full Installation'. It will download 'autoit-v3-setup.zip' file. Extract the .zip file to get 'autoit-v3-setup.exe' file

v. double click the .exe file and install the autoIT

vi. Upon installation we get following tools:

(a) Scite Script editor (This is the IDE where we have to write the autoIT code)

(b) autoIt window Info (This is to inspect the window elements properties to identify them)

(c) Run autoIt (This is used to runthe autoIT scripts)

(d) Compile script to .exe (This is used to convert the autoIt scripts (.au3) into .exe file

Q: Sample test case to automate notepad using autoIT:

1. Open notepad

2. Write some text in it

3. close the notepad, it will ask "Do you want to save...", click "Save" button

4. "Save as" window will open, enter file path & click on "Save" button

5. if "Confirm Save as" window appears, then click on "Yes" button

Q: How to integrate autoIT code with selenium webdriver?

Ans: 1. Write autoIt code using autiIT scite script editor & save it as <scriptName>.au3

2. Convert the .au3 file into .exe file using "Compile script to .exe" tool

3. execute the .exe file in selenium webdriver

Q: What might be the window components which occurs in web applications?

Ans: Upload, download, SSL certificates etc

Q: Test case to register to naukri using AutoIT:

1. Open browser and navigate to "https://www.naukri.com/"

2. click on "Register" link. It will open another child window. Switch to child window

3. Enter the required details viz., Full Name. Email ID, Password, Mobile number (Optional)

4. Click on "Upload Resume" button

5. Handle the Upload window component using autoIT

6. verify file name displayed successful

Sikuli: It is a independent functional testing tool. It supports both window and web applications.

It identifies the elements based on the images. Hence it is a image based tool.

As it is a image based tool, it is highly pixel sensitive.

It understands jython (java + Python).

Advantages:

1. It works for any kind of application.

Dis-advantags:

1. It is highly pixel sensitive

2. It doesnot work in a virtual computer. It requires active and physical computer.

3. for different browsers we need different images to be captured.

Q: How to download and install the sikuli?

Ans:1. Go to google and search for download sikuli

2. Open the sikuli official website (https://launchpad.net/sikuli/+download)

3. download sikulixide-2.0.5.jar (md5) -> SikuliX IDE for all systems

4. put this jar in a separate empty folder

5. Double click on the jar file, it will open the sikuli IDE

6. Write the scripts by taking the images. The extention for the sikuli script is .sikuli

Q: To write the scripts we need to use few sikuli methods?

Ans:

find(), type() click() doubleClick()

Q: The sikuli jar versions for sikuli IDE and sikuli integration with selenium?

Ans: sikulixide-2.0.5.jar: SikuliX IDE for all systems to write and run the scripts

sikulixapi-2.0.5.jar: SikuliX API for all systems to integrate with selenium webdriver

Q: How to integrate sikuli with selenium?

Ans: 1. Download sikuxaplii-2.0.5.jar, place it in eclipse std folder & set the build path

2. Copy the images and put it in eclipse std folder

3. Write a script using below classes

Screen

Match

Reflection:

It is a process of inspecting OR modifying the runtime behaviour of a class at runtime is k.a., Reflection.

Ex: methods, classes, interfaces, constructor

Note:

1. One advantage of reflection API in Java is, it can manipulate private members of the class too.

2. Most of the reflection members are checked exception. Hence try catch is required.

3. Reflection belongs to java.lang.reflect package

It can be achieved in 3 ways:

I. By creating object to the class. This is used for instance members

<className> <obj> = new <Constructor>();

II. by running static block alone

Class cls = Class.forName(<pkgName.className>);

III. By using Object class

Class cls = Class.forName(<pkgName.className>);

Object obj = cls.newInstance();

I. By creating instance to the class

<className> <obj> = new <Constructor>();

testNG:

testNG is a unit testing tool. TestNG is advanced than the Junit.

TestNG is an automation testing framework in which NG stands for "Next Generation".

TestNG is inspired by JUnit which uses the annotations (@).

TestNG overcomes the disadvantages of JUnit and is designed to make end-to-end testing easy.

It is mainly used to run the code in a systemmatic order based on the annotations present in the unit testing tools.

Key Difference between TestNG and JUnit

TestNG:

TestNG is a Java-based framework,

TestNG annotations are easier to use and understand than JUnit

TestNG allows us to create parallel tests,

In TestNG, Test cases can be grouped together.

JUnit:-

JUnit is an open-source Unit Testing Framework for JAVA.

JUnit does not support running parallel tests.

In JUnit, Grouping tests together is not possible.

It is a replacement for the main method.

Q: Advantages of TestNG over Junit?

1. It has more annotations than Junit

2. It has more conditions than Junit

3. In testng the execution can be controlled by using testng.xml file

4. TestNG has reporting mechanism (it automatically creates two different .html reports viz.,

(a) emailable-report.html

(b) index.html

5. DataProvider is present in testNG

6. Dependency testing can be achieved in testNG

7. We can group the tests and run them based on the group names.

8. In testNG batch execution is possible @ different levels through testng.xml file viz.,

(a) package level

(b) class level

(c) method level

8. All the annotaions names are not confusing like junit (Ex: @Before, @After). Here the annotaions names are straight forward.

9. Here methods can be either static OR non-static

Q: How to install testNG?

Case 1:

Go to Eclipse->Help->Eclipse MarketPlace-> search for testNG->once search is found & If the testNG is not installed then install it.

Case 2:

Go to Eclipse->Help->Install new Software->Add the URL https://testng.org/testng-eclipse-update-site -> Select testNG checkbox & continue installation.

Case 3:

Copy the TestNG jars (Manually download OR through Maven pom.xml you can download automatically) to eclipse & set the build path

TestNG annotations: Annotations in testNG are the built-in classes which are having a definite order of execution.

@Test

@BeforeSuite

@AfterSuite

@BeforeTest

@AfterTest

@BeforeClass

@AfterClass

@BeforeMethod

@AfterMethod

@BeforeGroups

@AfterGroups

@Parameters

@DataProvider

@Test: This annotation is used for Test scripts only. Here all the @Test will run in Ascending order based on the method name by default.

@BeforeSuite: The method with this annotation will always runs first with respect to testNG suite level.

@AfterSuite: The method with this annotation will always runs last with respect to testNG suite level

@BeforeTest: The method with this annotation will always runs first with respect to the test level which is present inside the testng.xml file

@AfterTest: The method with this annotation will always runs last with respect to the test level which is present inside the testng.xml file

@BeforeClass: The method with this annotation will always run first with respect to the perticular class.

@AfterClass: The method with this annotation will always run last with respect to the perticular class.

@BeforeMethod: The method with this annotation will always run first for every @Test annotations.

@AfterMethod: The method with this annotation will always run last for every @Test annotations

@BeforeGroups(): The method with this annotation will always run first with respect to the group name mentioned.

@AfterGroups(): The method with this annotation will always run last with respect to the group name mentioned.

@Parameters: This annotation is used to parameterize the testng @Test with simple data/values.

@DataProvider: This annotation is used to parameterize the testng Tests with complex data/values viz., Collection, Arrays, Object, data read from the property file, xml file, database etc.

Conditions: All conditions are associated with @Test annotation.

@Test(timeout=<milliseconds>)

@Test(expectedExceptions=<exception>.class)

@Test(enabled=false)

@Test(priority=<numbers>)

@Test(groups={"groups names"})

@Test(dataProvider={""})

@Test(dependsOnMethods={""})

@Test(dependsOnGroups={""})

@Test(alwaysRun=true)

@Test(invocationCount=<numbers>)

@Test(timeout=<milliseconds>): Timeout test in testng

@Test(expectedExceptions=<exception>.class): Exception test in testNG

@Test(enabled=false): The testNG will not execute the perticular test.

@Test(priority=<numbers>): We can prioritize the testNG tests & execute them based on the priority mentioned. The priority will start from zero/less than zero. If we don't mention the priority to the @Test, then by default the priority will be zero.

@Test(groups={"groups names"}): It is used to specify the group name to the testng tests

@Test(dataProvider={""}): It is used to use the dataprovide name to the perticular test

@Test(dependsOnMethods={""}): To achieve the dependency test at method level.

In dependency test, If the dependency method fails then the dependent method will be skipped by the testNG automatically

@Test(dependsOnGroups={""}): To achieve the dependency test at group level

In dependency test, If the depencecy group fails then the dependent group will be skipped by the testNG automatically

@Test(alwaysRun=true): In dependency test, If the dependency fails then the dependent will be skipped by the testNG automatically. To override this behaviour & to run the dependent test even after dependency fails we use this condition.

@Test(invocationCount=<numbers>): It is used to run the testNG tests Nth time

Q: Difference between Assert & Verify?

Assert: Assertion is a validation point.

Upon failure of Assertion, it will not run the below lines of code for the perticular block.

Verify: Verify is a validation point.

Upon failure of Verify, it will run the below lines of code for the perticular block as well. But overall test status will be failed.

Q: Assertions in Junit:

Ans:

Assert.assertTrue();

Assert.assertFalse();

Assert.assertNull();

Assert.assertNotNull();

Assert.assertEquals()

Assert.assertNotEquals();

Assert.assertSame();

Assert.assertNotSame();

Assert.assertArrayEquals(): It is not in testNG

Q: Assertions in testNG?

in testng 2 types of assertions are present

(1) Hard Assertion

Assert.assertTrue();

Assert.assertFalse();

Assert.assertNull();

Assert.assertNotNull();

Assert.assertEquals()

Assert.assertNotEquals();

Assert.assertSame();

Assert.assertNotSame();

(2) **Soft Assertion**

SoftAssert soft = new SoftAssert();

soft.assertTrue();

soft.assertFalse();

soft.assertNull();

soft.assertNotNull();

soft.assertEquals();

soft.assertNotEquals();

soft.assertSame();

soft.assertNotSame();

soft.assertAll()

Q: Difference between HardAssert & SoftAssert in testNG?

Ans:

HardAssert	SoftAssert
1. Upon failure, it will stop the execution of whole block OR below codes	1. Upon failure, it will not stop the block. Instead it will continue execution.
2. Results will be captured upon failure	2. Upon failure, it won't capture the result. Instead it shows test passed. To capture the failure we need to use .assertAll() method at the end of test
3. Assert class contains assertion methods which are static in nature	3. SoftAssert class contains assertion methods which are non-static

Q: Difference between .assertEquals() & .assertSame()?

Ans: **.assertEquals()**: It checks whether both the values are same

.assertSame(): It checks whether both the object references are same.

Q: What happens when priority=-1?

Q: What if 2 @Tests are having same priority?

Ex: both tests are having priority=1

Q: What If the mentioned group name doesnot exist?

Q: Can we achieve parallel execution using testNG?

Q: What happens If dependency test fails in testNG?

Q: What is the main purpose of testNG OR unit test tools?

Q: Advantages of testNG over Junit?

Q: Can we execute the java code without main method?

Q: Can we run the class members without creating a object to the class?

Q: Tell me the order of execution?

@Test

@BeforeMethod

@BeforeSuite

@AfterSuite

@BeforeGroups

@AfterTest

@AfterMethod

@AfterClass

Q: Perform a task multiple times (say 10 times) without using looping statements?

Q: How not to execute the tests in testNG?

Q: How to provide pre-condition & post-conditions to each test scripts in testNG?

Q: Different levels of executing the programmes in testNG?

Q: Can we give same name to both <test> in testng.xml file?

Q: How to fail the testNG tests?

Q: Types of assertions in testNG?

Q: Does SoftAssert captures the failures? How to capture it?

Q: The class is having methods with all the annotations except @Test? Will it get execute?

Q: How to re-execute the failed tests in testng?

Selenium waits

It is used to establish the sync between script speed and application rendering speed.

Types of waits:

In selenium there are 2 types of waits.

1. Implicit waits 2. Explicit waits

1**. Implicit waits:**

It is a hardcoded wait. The wait lasts for the given number of periods.

(a) pageLoadTimeout: Sets the amount of time to wait for a page load to complete before throwing an error.

(b) implicitlyWait: Specifies the amount of time the driver should wait when searching for an element if it is not immediately present.

(c) setScriptTimeout: Sets the amount of time to wait for an asynchronous script to finish execution before throwing an error.

2. Explicit waits:

These are the dynamic waits. Once the condition is achieved then it will skip waits for the remaing period of time.

Ex: Waiting for the element to load for 10 sec, but element loaded within 2 sec. Then dynamic waits will skip the remaining 8 sec.

(a) Thread.sleep()

(b) WebDriverWait class

(c) FluentWaits class

```
Wait<WebDriver> wait = new FluentWait<WebDriver>(oBrowser)
        .withTimeout(Duration.ofSeconds(40))
        .pollingEvery(Duration.ofSeconds(10))
        .ignoring(NoSuchElementException.class);
wait.until(ExpectedConditions.elementToBeClickable(By.xpath("")));
wait.until(ExpectedConditions.textToBePresentInElementLocated(By.xpath(""), "<text>"));
```

Common Exceptions in Selenium WebDriver:

Selenium has its own set of exceptions. While developing selenium scripts, a programmer has to handle or throw those exceptions.

Note: All runtime exception classes in Selenium WebDriver come under the superclass WebDriverException.

Though there are many Exception classes under WebDriverException, we commonly see the below ones:

(1) NoSuchElementException: This commonly seen exception class, it is a subclass of NotFoundException class. The exception occurs when WebDriver is unable to find and locate elements.

(2) NoSuchWindowException: NoSuchWindowException comes under NotFoundException class. This is thrown when WebDriver tries to switch to an invalid window.

(3) NoSuchFrameException: When WebDriver is trying to switch to an invalid frame, NoSuchFrameException under NotFoundException class is thrown.

(4) NoAlertPresentException: NoAlertPresentException under NotFoundException is thrown when WebDriver tries to switch to an alert, which is not available.

org.openqa.selenium.NoAlertPresentException will be thrown If below automation code calls accept() operation on Alert() class when an alert is not yet on the screen.

(5) InvalidSelectorException: This subclass of NoSuchElementException class occurs when a selector is incorrect or syntactically invalid. This exception occurs commonly when XPATH locator is used.

Ex: oBrowser.findElement(By.xpath("//button[@type='button'][100]));

This would throw an InvalidSelectorExeption because the XPATH syntax is incorrect.

(6) ElementNotVisibleException: ElementNotVisibleException class is a subclass of ElementNotInteractableException class. This exception is thrown when WebDriver tries to perform an action on an invisible web element, which cannot be interacted with. That is, the web element is in a hidden state.

(7) ElementNotSelectableException: This exception comes under InvalidElementStateException class. ElementNotSelectableException indicates that the web element is present in the web page but cannot be selected.

For example, the below code can throw a ElementNotSelectableException if the id "swift" is disabled.

Select dropdown = new Select(driver.findElement(By.id("swift")));

(8) TimeoutException: This exception occurs when a command completion takes more than the wait time. Waits are mainly used in WebDriver to avoid the exception ElementNotVisibleException.

Sometimes test page might not load completely before next command in the program. If WebDriver tries to find an element in the webpage before the page completely loads, then exception ElementNotVisibleException is thrown. To avoid this exception, waits commands are added.

However, if the components don't load even after the wait, the exception org.openqa.selenium.TimeoutException will be thrown.

(9) NoSuchSessionException: This exception is thrown when a method is called after quitting the browser by WebDriver.quit(). This can also happen due to web browser issues like crashes and WebDriver cannot execute any command using the driver instance.

To see this exception, the code below can be executed.

driver.quit()

Select dropdown = new Select(driver.findElement(By.id("swift")));

*****(10) StaleElementReferenceException:** This exception says that a web element is no longer present in the web page.

This error is not the same as ElementNotVisibleException.

StaleElementReferenceException is thrown when an object for a particular web element was created in the program without any problem and however; this element is no longer present in the window. This can happen if there was a

Navigation to another page

DOM has refreshed

A frame or window switch

```
WebElement firstName = driver.findElement(By.id("firstname"));

driver.switchTo().window(Child_Window);

firstName.sendKeys("Aaron");
```

In the code above, object firstName was created and then the window was switched. Then, WebDriver tries to type "Aaron" in the form field. In this case StaleElementReferenceException is thrown.

(11) NoSuchAttributeException: While trying to get attribute value but the attribute is not available in DOM.

(12) WebDriverException: Exception comes when a code is unable to initialize WebDriver.

---------List of all common exceptions in Selenium---------

Exceptions

(1) ElementClickInterceptedException(msg, …) : The Element Click command could not be completed because the element receiving the events is obscuring the element that was requested to be clicked.

(2) ElementNotInteractableException(msg, screen, …) : Thrown when an element is present in the DOM but interactions with that element will hit another element due to paint order.

(3) ElementNotSelectableException(msg, screen, …) : Thrown when trying to select an unselectable element.

(4) ElementNotVisibleException(msg, screen, …): Thrown when an element is present on the DOM, but it is not visible, and so is not able to be interacted with.

(5) ImeActivationFailedException(msg, screen, …): Thrown when activating an IME engine has failed.

(6) ImeNotAvailableException(msg, screen, stacktrace) : Thrown when IME support is not available.

(7) InsecureCertificateException(msg, screen, …) : Navigation caused the user agent to hit a certificate warning, which is usually the result of an expired or invalid TLS certificate.

(8) InvalidArgumentException(msg, screen, stacktrace) : The arguments passed to a command are either invalid or malformed.

(9) InvalidCookieDomainException(msg, screen, …): Thrown when attempting to add a cookie under a different domain than the current URL.

(10) InvalidCoordinatesException(msg, screen, …): The coordinates provided to an interaction's operation are invalid.

(11) InvalidElementStateException(msg, screen, …): Thrown when a command could not be completed because the element is in an invalid state.

(12) InvalidSelectorException(msg, screen, stacktrace) : Thrown when the selector which is used to find an element does not return a WebElement.

(13) InvalidSessionIdException(msg, screen, …): Occurs if the given session id is not in the list of active sessions, meaning the session either does not exist or that it's not active.

(14) InvalidSwitchToTargetException(msg, screen, …): Thrown when frame or window target to be switched doesn't exist.

(15) JavascriptException(msg, screen, stacktrace): An error occurred while executing JavaScript supplied by the user.

(16) MoveTargetOutOfBoundsException(msg, screen, …): Thrown when the target provided to the ActionsChains move() method is invalid.

(17) NoAlertPresentException(msg, screen, stacktrace) : Thrown when switching to no presented alert.

(18) NoSuchAttributeException(msg, screen, stacktrace): Thrown when the attribute of element could not be found.

(19) NoSuchCookieException(msg, screen, stacktrace): No cookie matching the given path name was found amongst the associated cookies of the current browsing context's active document.

(20) NoSuchElementException(msg, screen, stacktrace): Thrown when element could not be found.

(21) NoSuchFrameException(msg, screen, stacktrace) : Thrown when frame target to be switched doesn't exist.

(22) NoSuchShadowRootException(msg, screen, …): Thrown when trying to access the shadow root of an element when it does not have a shadow root attached.

(23) NoSuchWindowException(msg, screen, stacktrace): Thrown when window target to be switched doesn't exist.

(24) ScreenshotException(msg, screen, stacktrace) : A screen capture was made impossible.

(25) SessionNotCreatedException(msg, screen, …): A new session could not be created.

(26) StaleElementReferenceException(msg, screen, …) : Thrown when a reference to an element is now "stale".

(27) TimeoutException(msg, screen, stacktrace): Thrown when a command does not complete in enough time.

(28) UnableToSetCookieException(msg, screen, …): Thrown when a driver fails to set a cookie.

(29) UnexpectedAlertPresentException(msg, screen, …): Thrown when an unexpected alert has appeared.

(30) UnexpectedTagNameException(msg, screen, …): Thrown when a support class did not get an expected web element.

(31) UnknownMethodException(msg, screen, stacktrace): The requested command matched a known URL but did not match any methods for that URL.

(32) WebDriverException(msg, screen, stacktrace): Base webdriver exception.

Java Script

Javascript is a scripting language. It is mainly used for client side validations.

The javascript will be written directly inside the html OR externally under .js files.

In html DOM, we use <script> tag to write the javascript. The <script> tag can be used inside <head> tag OR <body> tag.

```
<script>
  <!-- Input & output statements in javascript-->
  var strName = prompt("Enter Name");
  alert("Name is: "+strName);
  window.alert("Name is: "+strName);
  document.write("Name is: "+strName);
  console.log("Name is: "+strName);
</script>
```

****

```
<script>
```

```
  <!-- Variable declaration-->
  var strName;
  strName = "Kalam";
  document.write("Name: "+strName+"<br>");
  var strCity = "Raichur";
  document.write("City: "+strCity+"<br>");
  var sName, sCity, sAddress;
  sName = "Modi";
  sCity = "Gujarath";
  sAddress = "Indian";
  document.write("sName: "+sName+"<br>");
  document.write("sCity: "+sCity+"<br>");
  document.write("sAddress: "+sAddress+"<br>");
</script>
```

****

```
<script>
  <!-- Datatypes in JavaScript
  //String
  //boolean
  //int
  //object
  //array
  -->
```

```
   var sName = "Modi";
   var blnRes = true;
   var age = 30;
   //Object in javascript
   var obj = {FN:"Abdul", LN:"Kalam", age:70};
   var sFN = obj.FN;
   var sLN = obj.LN;
   var iAge = obj.age;
   document.write(sFN+" "+sLN+" "+iAge+"<br>");
   document.write("**************"+"<br>");
   for(x in obj)
   {
         document.write(" "+obj[x]+"<br>");
   }
</script>
**********************************************
<script>
   <!-- Static Array in JavaScript-->
   var arr = ["A","B","C","D"];
   document.write("arr[0] = "+arr[0]+"<br>");
   document.write("arr[1] = "+arr[1]+"<br>");
   document.write("arr[2] = "+arr[2]+"<br>");
   document.write("arr[3] = "+arr[3]+"<br>");
```

```
document.write("*****************"+"<br>");
for(var i=0;i<arr.length;i++)
{
	document.write("arr["+i+"] = "+arr[i]+"<br>");
}
document.write("*****************"+"<br>");
<!-- Dynamic Array in JavaScript-->
var arr1 = {};
arr1[0] = "Apple";
arr1[1] = true;
arr1[2] = 100;
for(x in arr1)
{
	document.write("for each: "+arr1[x]+"<br>");
}
document.write("*****************"+"<br>");
//add value during runTime
var arr2 = ["Apple","Boy"];
//.push(): to add value to array during runTime
arr2.push("Sunday");
arr2.push("Monday");
arr2.push("Tuesday");
arr2.push("Wednesday");
```

```
for(x in arr2)
{
	document.write("Adding value: "+arr2[x]+"<br>");
}
document.write("*****************"+"<br>");
//.pop(): to delete the last element in array
arr2.pop();
for(x in arr2)
{
	document.write("Delete value: "+arr2[x]+"<br>");
}
document.write("*****************"+"<br>");
//to delete the required element in array
delete arr2[0];
for(x in arr2)
{
	document.write("Delete value: "+arr2[x]+"<br>");
}
document.write("*****************"+"<br>");
//To convert array to String
var kk = arr2.join(" ");
document.write(kk);
</script>
```

****

```
<script>
  <!-- Looping statements-->
  //for loop
  for(var i=0;i<=10;i++)
  {
       if((i%2)==0)
       {
            document.write("Even Number's: "+i+"<br>");
       }
  }
  document.write("***************"+"<br>");
  //for each loop
  var arr = [100,200,300];
  for(z in arr)
  {
       document.write("Array Value: "+arr[z]+"<br>");
  }
  document.write("***************"+"<br>");
  //while loop
  var k=0;
  while(k<=5)
  {
```

```
        document.write("While: "+k+"<br>");
        k++;
    }
    document.write("***************"+"<br>");
    //do while loop
    var m=0;
    do
    {
        document.write("Do While: "+m+"<br>");
        m++;
    }while(m<=5);
</script>
**************************************
<script>
    <!-- Conditional statements-->
    var age = 21;
    if((age>=21) && (age<=100))
    {
        document.write("Eligible for voting <br>");
    }else{
        document.write("Not Eligible for voting <br>");
    }
    document.write("*****************<br>");
```

```
  var sColor = "blue";
  switch(sColor.toLowerCase())
  {
        case "red":
              document.write("Danger <br>");
              break;
        case "white":
              document.write("Clean <br>");
              break;
        case "blue":
              document.write("Ocean <br>");
              break;
        default:
              document.write("Invalid");
  }
</script>
**********************************
<script>
  //Operators in JavaScript
  I. ArithMatic Operators
  +, -, *, /, %
  II. Comparision Operators
  ==, ===, >, >=, <, <=
```

III. Logical Operators

&&, ||, !

IV. Assignment Operators

=

V. Concatination Operator

+

VII. Increment

i++, i=i+1

VIII. Decrement

i--, i=i-1

```
</script>
**************************************
<script>
  //Difference betn == & ===
  var sCity1 = "Raichur";
  document.write("sCity1 Type is: "+typeof(sCity1)+"<br>");
  var sCity2 = new String("Raichur");
  document.write("sCity2 Type is: "+typeof(sCity2)+"<br>");
  if(sCity1==sCity2)
  {
      document.write("Both value same <br>");
  }else{
      document.write("Both value NOT same <br>");
```

```
  }
  if(sCity1===sCity2)
  {
        document.write("Both value&Type same <br>");
  }else{
        document.write("Both value&Type NOT same <br>");
  }
</script>
```

****

```
<script>
  //try catch
  try{
        alert("Sample alert");
  }catch(err)
  {
       document.write("Error: "+err);
  }
</script>
```

****

```
<script>
  //String methods
  var str = "India is great";
  document.write("Upper: "+str.toUpperCase()+"<br>");
```

```
   document.write("Lower: "+str.toLowerCase()+"<br>");
   document.write("Extract: "+str.substr(0,5)+"<br>");
</script>
**********************************************
//Functions and Events in javascript
<html>
  <head><title>JavaScript Demo</title>
    <script>
          function display()
          {
            document.write("display method <br>");
          }
          function addition(a,b)
          {
             var result = a+b;
                 document.write("Total "+result+" <br>");
          }
          function multiplication()
          {
             return 10*2;
          }
          function test()
          {
```

```
            var returnVal = multiplication();
            document.write("returnVal: "+returnVal+"<br>");
          }
        </script>
  </head>
  <body>
    <p id="para1">sample Paragraph1</p>
        <p id="para2">sample Paragraph2</p>
    <input type="text" id="id1" name="name1"></input><br><br>
        <input type="text" id="id1" name="name2"></input><br><br>
        <input type="text" id="id2" name="name1"></input><br><br>
        <input type="text" id="id3" name="name1"></input><br><br>
        <input type="text" id="id4" name="name3" class="class2"></input><br><br>
        <input type="text" id="id5" name="name4" class="class2"></input><br><br>

        <input type="button" id="btn_id1" name="btn_name1" class="btn_cls1" value="Click Here1" onclick="addition(10,20)"></input>

        <input type="button" id="btn_id1" name="btn_name1" class="btn_cls1" value="Click Here2" onclick="test()"></input>
  </body>
```

```
</html>
**********************************************
<script>
  //Read the text
  function readValue()
  {
        var kk1 = document.getElementById('para1').innerHTML;
        alert(kk1);
        var kk2 = document.getElementById('para2').innerHTML;
        document.write(kk2);
  }
</script>
**********************************************


<script>
  //Update the text
  function readValue()
  {
      var kk1 = document.getElementById('para1').innerHTML;
      alert(kk1);
      document.getElementById('para1').innerHTML="Updated "+kk1
  }
</script>
```

****

```
<script>
  //Apply CSS (Style)
  function readValue()
  {
       document.getElementById('para1').style.color="red";
  }
</script>
```

****

```
<script>
  //get the element attribute
  function readValue()
  {
       var kk = document.getElementById('btn_id1').getAttribute("class");
       document.write(kk);
  }
</script>
```

****

```
<script>
  //Set the element attribute
  function readValue()
```

```
  {
document.getElementById('btn_id1').setAttribute("class","AAAAAAA
AAA");
        var kk1 =
document.getElementById('btn_id1').getAttribute("class");
        alert(kk1);
  }
</script>
```

****

Q: How to find the elements in DOM staructure?

Ans:

.getElementById()

.getElementsByClassName()

.getElementsByName()

.getElementsByTagName()

.querySelectorAll()

html object model: document

```
<script>
  //find the element by ID
  function readValue()
  {
        var kk = document.getElementById('id4');
        kk.value = "ADMIN";
  }
```

```
</script>
**********************************************
<script>
  //find the element by ClassName
  function readValue()
  {
        var kk = document.getElementsByClassName('class2');
        alert(kk.length);
        kk[0].value="AAAAAA";
        //clear the value
        kk[0].value="";
        kk[1].value = "BBBBBBB";
        //clear the value
        kk[1].value = "";
        var i=0;
        for(x in kk)
        {
              kk[x].value = "Value"+(i+1);
              i++;
        }
  }
</script>
**********************************************
```

```
<script>
  //find the element by TagName
  function readValue()
  {
        var kk = document.getElementsByTagName('input');
        alert(kk.length);
        var i=0;
        for(x in kk)
        {
              var sType = kk[x].getAttribute("type");
              if(sType!=="button")
              {
                    kk[x].value = "Value"+(i+1);
                    i++;
              }
        }
  }
</script>
```

****

```
<script>
  //find the element by Name
  function readValue()
  {
```

```
        var kk = document.getElementsByName('name1');
        alert(kk.length);
        var i=0;
        for(x in kk)
        {
            kk[x].value = "Value"+(i+1);
            i++;
        }
  }
</script>
```

****

```
<script>
      //find the element by cssSelector
      function readValue()
      {
        var kk = document.querySelectorAll('input#id1');
        alert(kk.length);
        var i=0;
        for(x in kk)
        {
            kk[x].value = "Value"+(i+1);
            i++;
        }
```

```
        }
</script>
```

****

```
<script>
  //find the element by html object model
  function readValue()
  {
        var url = document.URL;
        alert(url);
        var title = document.title;
        alert(title);
        var intLinks = document.links;
        alert("Links: "+intLinks.length);
        var intForms = document.forms;
        alert("Forms: "+intForms.length);
  }
</script>
```

****

JavaScript in Selenium WebDriver

```
JavascriptExecutor js = (JavascriptExecutor) oBrowser;
//Read URL
String strURL = (String) js.executeScript("var kk=document.URL; return kk");
```

```
System.out.println(strURL);

//Read Title

String strTitle = (String) js.executeScript("var kk=document.title; return kk");

System.out.println(strTitle);

//# of links

long iLinks = (long) js.executeScript("var kk=document.links; return kk.length");

System.out.println(iLinks);
```

****

```
JavascriptExecutor js = (JavascriptExecutor) oBrowser;

//Enter the value in text fields

js.executeScript("document.getElementById('id2').value='aaaaaa';");

js.executeScript("var kk = document.getElementById('id1'); kk.value='bbbbbbbb';");

WebElement oXpath = oBrowser.findElement(By.xpath("//input[@id='id4']"));

js.executeScript("arguments[0].value='CCCCC';", oXpath);
```

****

```
JavascriptExecutor js = (JavascriptExecutor) oBrowser;

//Enter the value in text fields using className

js.executeScript("var kk = document.getElementsByClassName('class2'); kk[0].value='AAAAA';");
```

```
List<WebElement> oXpath = oBrowser.findElements(By.className("class2"));

js.executeScript("arguments[0].value='CCCCC';", oXpath.get(1));

*****************************************

JavascriptExecutor js = (JavascriptExecutor) oBrowser;

//Enter the value in text fields using TagName

js.executeScript("var kk = document.getElementsByTagName('input'); kk[0].value='AAAAA';");

List<WebElement> oXpath = oBrowser.findElements(By.tagName("input"));

js.executeScript("arguments[0].value='CCCCC';", oXpath.get(1));

*********************************************

JavascriptExecutor js = (JavascriptExecutor) oBrowser;

//Enter the value in text fields using Name

js.executeScript("var kk = document.getElementsByName('name1'); kk[0].value='AAAAA';");

List<WebElement> oXpath = oBrowser.findElements(By.name("name2"));

js.executeScript("arguments[0].value='CCCCC';", oXpath.get(0));

***********************************

JavascriptExecutor js = (JavascriptExecutor) oBrowser;

//Enter the value in text fields using cssSelector

js.executeScript("var kk = document.querySelectorAll('#id1'); kk[0].value='AAAAA';");
```

```
List<WebElement> oXpath = oBrowser.findElements(By.cssSelector("#id2"));

js.executeScript("arguments[0].value='CCCCC';", oXpath.get(0));

**************************************

JavascriptExecutor js = (JavascriptExecutor) oBrowser;

//click the first links

js.executeScript("var kk = document.getElementsByTagName('a'); kk[0].click();");

List<WebElement> oXpath = oBrowser.findElements(By.tagName("a"));

js.executeScript("arguments[0].click();", oXpath.get(1));

******************************************

//select the dropdown value using value attribute

JavascriptExecutor js = (JavascriptExecutor) oBrowser;

js.executeScript("document.getElementById('list1').value='blore';");

//Another way to select drop down value

String script="var kk = document.getElementById('list2');";

script+="for(var i=0;i<kk.length;i++){;";

script+="if(kk.options[i].text=='Raichur'){;";

script+="kk.options[i].selected=1";

script+="}";

script+="}";

js.executeScript(script);
```

System.out.println("**********Read URL**********");

String strURL=(String) js.executeScript("var kk=document.URL; return kk");

System.out.println(strURL);

System.out.println("**********Read Tilte**********");

String strTitle=(String) js.executeScript("var kk=document.title; return kk");

System.out.println(strTitle);

System.out.println("**********Enter value in the text fileds**********");

#//case 1:

JavascriptExecutor js=(JavascriptExecutor) oBrowser;

js.executeScript("document.getElementById('edit_id1').value='username 1';");

#//case 2:

js.executeScript("document.getElementById('edit_id2').value=argument s[0];","username2");

#ClassName

JavascriptExecutor js=(JavascriptExecutor) oBrowser;

long intEdit1=(long) js.executeScript("var kk=document.getElementsByClassName('edit_class1'); return kk.length");

System.out.println(intEdit1);

js.executeScript("var kk=document.getElementsByClassName('edit_class1'); kk[1].value='ggggggggg';");

#TagName

long intEdit2=(long) js.executeScript("var kk=document.getElementsByTagName('input'); return kk.length");

System.out.println(intEdit2);

js.executeScript("var kk=document.getElementsByTagName('input'); kk[3].value='thirdOne';");

#cssSelector

long intEdit3=(long) js.executeScript("var kk=document.querySelectorAll('input#edit_id4'); return kk.length");

System.out.println(intEdit3);

js.executeScript("var kk=document.querySelectorAll('input#edit_id4'); kk[0].value='CssSelector';");

#Xpath or WebElement in javascript

JavascriptExecutor js=(JavascriptExecutor) oBrowser;

WebElement oEdit=oBrowser.findElement(By.xpath("//input[@id='edit_id2']"));

js.executeScript("arguments[0].value='sampleTest';",oEdit);

//==========

List<WebElement> oText=oBrowser.findElements(By.tagName("input"));

js.executeScript("arguments[0].value='sampleTest';",oText.get(5));

//Select value from the list

JavascriptExecutor js=(JavascriptExecutor) oBrowser;

String script="var kk=document.getElementById('list1');";

script+="for(var i=0;i<kk.length;i++){;";

```
script+="if(kk.options[i].text=='Raichur'){;";

script+="kk.options[i].selected='1';";

script+="}";

script+="}";

js.executeScript(script);
```

$$$$$$$$ JavaScript $$$$$$$$

```
//read the URL

JavascriptExecutor js = (JavascriptExecutor) oBrowser;

String sURL = (String) js.executeScript("var url = document.URL; return url");

System.out.println(sURL);

//Read the title

JavascriptExecutor js = (JavascriptExecutor) oBrowser;

String sTitle= (String) js.executeScript("var title = document.title; return title");

System.out.println(sTitle);

//Find links count & name

JavascriptExecutor js = (JavascriptExecutor) oBrowser;

long count = (long) js.executeScript("var kk = document.getElementsByTagName('a'); return kk.length");

System.out.println(count);

String sName = (String) js.executeScript("var kk = document.getElementsByTagName('a'); return kk[0].innerHTML;");

System.out.println(sName);
```

```
//click the element

JavascriptExecutor js = (JavascriptExecutor) oBrowser;

js.executeScript("document.getElementById('links').click();");

//enter value

JavascriptExecutor js = (JavascriptExecutor) oBrowser;

js.executeScript("document.getElementById('edit_id1').value='Abdul';")
;

js.executeScript("document.getElementById('edit_id2').value=argument
s[0];","Kalam");

WebElement oEle =
oBrowser.findElement(By.xpath("//input[@id='edit_id3']"));

js.executeScript("arguments[0].value='Raichur';", oEle);

//getElementsByName

JavascriptExecutor js = (JavascriptExecutor) oBrowser;

js.executeScript("var kk =
document.getElementsByName('edit_name1');
kk[3].value='Bangalore';");

//getElementsByClassName

JavascriptExecutor js = (JavascriptExecutor) oBrowser;

js.executeScript("var kk =
document.getElementsByClassName('edit_class1');
kk[3].value='Bangalore';");

//querySelectorAll

JavascriptExecutor js = (JavascriptExecutor) oBrowser;
```

```
js.executeScript("var kk=document.querySelectorAll('#edit_id1'); kk[0].value='admin';");

//drop down values

JavascriptExecutor js = (JavascriptExecutor) oBrowser;

String script="var kk = document.getElementById('list1');";

script+="for(var i=0;i<kk.length;i++){;";

script+="if(kk.options[i].text=='Sindhanur'){;";

script+="kk.options[i].selected=1";

script+="}";

script+="}";

js.executeScript(script);

//++++++++++++++++++++++++++++++++++++++++

Thread.sleep(2000);

js.executeScript("var kk=document.getElementById('list1'); kk.value='rcr';");

JavascriptExecutor js=(JavascriptExecutor) oBrowser;

js.executeScript("scroll(0,250)");       //scroll down

Thread.sleep(2000);

js.executeScript("scroll(0,-250)");      //scroll up

==============================================

JavascriptExecutor js = (JavascriptExecutor ) oBrowser;

//Scroll the page 400 pixel down

js.executeScript("scroll(0, 400)");
```

```
 //Scroll the page 400 pixel up
js.executeScript("scroll(0, -400)");
//Scroll the page 400 pixel to the right
js.executeScript("scroll(400, 0)");
//Scroll the page 400 pixel to the left
js.executeScript("scroll(-400, 0)");
//Scroll the WebElement into view
Webelement element = oBrowser.findElement(By.id("username"));
js.executeScript("arguments[0].scrollIntoView();", element);
//Scroll to the bottom of the page
js.executeScript("window.scrollTo(0, document.body.scrollHeight)");
//Scroll to the top of the page
js.executeScript("window.scrollTo(document.body.scrollHeight, 0)");
```

POM: Page Object Model

Page Object Model is a design pattern to create Object Repository for web UI elements. Under this model, for each web page in the application, there should be corresponding page class. This Page class will find the WebElements of that web page and also contains Page methods which perform operations on those WebElements.

The POM is to enhance the reusability, hence we are creating the Page Factory for all the pages OR modules OR scenarios.

The each Page Factory consist of Locators (WebElement), methods which belongs to that perticular page. Likewise create a

multiple Page factories and use them in your framework to create the test scripts.

To create elements in POM we use following annotations:

@FindBy(<locators>)

@FindBys(<locators>)

@CacheLookUp

@FindBy: Is used to define the locator within the page factory.

@FindBys: Is used to define the list of matching locators within the page factory.

@CacheLookUp: It is used for the static WebElement. If the element is created once then it will be cached & next time If you want to use that element, then it will be taken from the cache directly.

This @CacheLookUp annotation must be followed by @FindBy annotation

Advantages of POM

1. Page Object Design Pattern says operations and flows in the UI should be separated from verification. This concept makes our code cleaner and easy to understand.

2. The Second benefit is the object repository is independent of test cases, so we can use the same object repository for a different purpose with different tools. For example, we can integrate Page Object Model in Selenium with TestNG/JUnit for functional Testing and at the same time with JBehave/Cucumber for acceptance testing.

3. Code becomes less and optimized because of the reusable page methods in the POM classes.

4. Methods get more realistic names which can be easily mapped with the operation happening in UI. i.e. if after clicking on the button we land on the home page, the method name will be like ‘gotoHomePage()’.

AjaxElementLocatorFactory

AjaxElementLocatorFactory is a lazy loading concept of PageFactory in Selenium. It is used to find the web elements only when the elements are used in any operation. It assigns a timeout for WebElements to the object page class. One of the key advantages of using the pattern PageFactory in Selenium is AjaxElementLocatorFactory Class.

Here, when an operation is performed on an element the wait for its visibility starts from that moment only. If the element is not found in the given time interval, Test Case execution will throw 'NoSuchElementException' exception.

```
AjaxElementLocatorFactory factory = new AjaxElementLocatorFactory(oBrowser, 100);

PageFactory.initElements(factory, this);
```

Build tools:

Modern software development relies heavily on build tools that facilitate the transformation of source code into deployed software.

Ex: Java program-->.java-->.class-->.jar

Many build tools are available in market viz., ANT, Maven, Gradle, Ivy, Grapes etc.

Here we are stydying Maven as build tool.

Uses of maven:

1. It provides a simple project setup for the developers.

2. Maven creates outstanding dependency management for your projects (download the JAR files of your projects from the central repository).

3. We can integrate third party tools using as plugins

4. Increases reusability

5. Reduces extra steps like creating jar/war/ear files, building reports, or executing Junit/testNG test cases.

Using maven tool we can create the maven project and we can use it.

Q: How to download and install the maven

1. Go to google and search "Download Apache Maven"

2. Click and open the first official link

3. download the .zip file (windows) or tar.tz (for mac/Linux/unix)

4. Extract the .zip file

5. Set the class path for maven with the name as MAVEN_HOME/M2_HOME

6. check maven is installed properly by running below command in command prompt:

mvn -version

Q: How to create the maven project?

Ans: there are 2 ways we can create the maven project

(i) from eclipse/IDE:

go to eclipse-->File-->New-->Project-->Search for maven-->select 'Maven Project'-->click 'Next' button-->click 'Next' button-->filter by "maven-archetype-quickstart"-->select the one whose group id is 'maven-archetype-quickstart' and click 'Next' -->Provide groupId (package name) & artifactId (Project name)-->click on 'Finish' button.

(ii) using command prompt

Create a workspace-->Open the command prompt and navigate to the workspace created using command prompt-->run the below commands:

mvn archetype:generate

.

the execution pauses at below point:

Choose a number OR apply filter (Maven-Archetype-Quickstart):

Press ENTER

choose the version & press Enter

.

groupId (Package name)

ArtifactID (project name)

Press ENTER

.

.Y

BUILD SUCCESS

Note: If you create the project from command prompt, then you need to import that maven project into eclipse.

Q: How to import the maven project into eclipse?

Ans: Go to eclipse-->File-->Import-->search Maven & select 'Existing Maven Project'-->click 'Next'--->seach and select the workspace in the Root Directory, it will load the maven project automatically-->click on 'Finish' button

Q: Why build tools are required?

Ans: 1. It will create the project structuresas per the need.

2. It will compile and check the health of your code/project periodically. If any mistake the build will fail.

Q: What is the maven central repo url?

Ans: https://repo.maven.apache.org/

Q: In local machine where the maven dowmloads the dependencies?

Ans: Maven will download the artifacts into maven local repository. Maven will create a .m2 folder under "C:\Users\<userName>\.m2\repository"

Q: How to know the folder structure of the jars in maven central repository?

Ans: Go to https://mvnrepository.com/ website---> search artifacts what you want to download. Select the suitable version. It will provide you the dependency.

Q: Maven Life Cycle?

Ans:

mvn clean: it will clean the project. It will delete the /target folder which includes class files and jar files

mvn compile: It will compile all the source files & create the class files

mvn clean install: It will clean, compile and it will execute the maven project

mvn eclipse:eclipse: It will convert the maven project into eclipse project. Which means it will creates .project and .class files and set the build path for all your jar files.

mvn test: It will run the maven project.

mvn package: It will build the maven project and creates the .jar file

Available lifecycle phases are:

pre-clean, clean, post-clean, validate, initialize, generate-sources, process-sources, generate-resources, process-resources, compile, process-classes, generate-test-sources, process-test-sources, generate-

test-resources, process-test-resources, test-compile, process-test-classes, test, prepare-package, package, pre-integration-test, integration-test, post-integration-test, verify, install, deploy, pre-site, site, post-site, site-deploy.

Automation Life Cycle:

1. Identify the need for automation

|

V

2. Get customer approval

|

V

3. Perform tool Evaluation

In interview tell that the tool was already standardized in our organization.

[Ex:

Selenium, UFT, Test Complete, Sahi, Watir, Tosca,

Kalalon, Cypress, Browser stack, PlayWright etc]

|

V

4. Identify the Critical Regression test cases & Create a First Draft of the framework

|

V

5. Setup a meeting for POC (Proof Of Concept). All the automation people, BA, Managers etc from the division will attend the meeting

|

V

6. Evaluate the review comments & accept the required ones & fix the review comments in your framework.

|

V

7. Place the framework into the git repository & ask the remaing people to take the framework form the repository (clone the repository)

|

V

8. Identify the Regression test cases which can be automated and add them to the automation Suite

|

V

9. Convert the identified regression test cases into Automation test scripts by writing methods & test scripts.

|

10. After completion of coding, Perform dry run and make sure the code is working as expected + All the coding standards are followed.

Send for code review through version control tools [pull Request(PR)] (Ex: Git, SVN, perforse etc)

|

V

11. The reviewer will provide the review comments If required.

Fix all the review comments and create a pull Request(PR). If every thing is fine, then your code will me merged into main branch in the verison control tool.

|

V

12. Whenever build comes, checkout the framework from the version control tool & execute the framework against the each builds

| (Execution flow will be followed)

V

13. Create a consolidated Automation test execution report & share with all stake holders

BDD: Behavior Driven Development:

Cucumber Framework is a key tool in Behavior Driven Development (BDD), known for bridging the gap between technical teams and business stakeholders. It operates on executable specifications written in Gherkin, a plain language accessible to all team members, fostering inclusivity and collaboration.

Its emphasis on clear communication minimizes misunderstandings, streamlining the development process. Its compatibility with multiple programming languages like Java, Ruby, and .Net further enhances its adaptability in various software development settings.

How to Start with Cucumber Testing?

1. Understand the Basics of BDD and Gherkin:

Before diving into Cucumber, familiarize yourself with Behavior-Driven Development (BDD) principles and the Gherkin language. This foundational knowledge is crucial for the effective use of Cucumber.

2. Choose a Programming Language:

Decide on the programming language you will use for writing your step definitions. Cucumber supports several languages, including Java, Ruby, and .NET.

3. Install Cucumber:

Install Cucumber on your system. This process will vary depending on your chosen programming language and development environment.

4. Set Up Your First Feature File:

Create a feature file in Gherkin. This file will contain your first test scenario, described in a format that Cucumber can understand.

5. Write Step Definitions:

For each step in your Gherkin scenario, write corresponding step definitions. These are the actual code snippets that will execute the steps of your test.

6. Run Your Test:

Execute the test to see if your scenario passes or fails. This will validate whether the software behaves as expected according to your Gherkin scenario.

7. Iterate and Expand:

Continue adding more feature files and step definitions, expanding your test coverage and refining your testing process.

Here is the list of keywords that Gherkin supports:

1. Feature: Feature defines the logical test functionality you will test in this feature file

2. Background: Background is used to define steps that are common to all the tests in the feature file

3. Scenario: Each Feature will contain a number of tests to test the feature. Each test is called a Scenario and is described using the Scenario: keyword.

4. Scenario Outline: In situations where one wants to execute the same Scenario with various combinations of values or arguments, one could use the Scenario Outline. It facilitates the testing of the same scenario with multiple arguments. The multiple arguments passed against the same scenario outline are called Examples which is another keyword in Cucumber

5. Given: Given defines a precondition to the test.

ex: Scenario: Successful LogIn with Valid Credentials

6. When: When keyword defines the test action that will be executed. By test action we mean the user input action.

7. Then: Then keyword defines the Outcome of previous steps

8. And: And keyword is used to add conditions to your steps

9. But: But keyword is used to add negative type comments. It is not a hard & fast rule to use but only for negative conditions

10. '*'

Feature:

Given User is on Home Page

And Login Link is displayed

When User Navigates to Login form

And User enters email and Password

Then Login Successfully will be displayed

And Logout Link should be displayed

Q: How to setup BDD cucumber:

Ans: Add the below dependencies in the pom.xml

these cucumber belongs to io.cucumber package.

1. cucumber-java
2. cucumber-core
3. cucumber-testNG
4. gherkin

add the plugins also

cucumber++

gherkin

cucumber-java

Advantages of BDD with Cucumber:

Collaboration: BDD encourages collaboration between developers, QA, and business stakeholders since scenarios are written in plain language that everyone can understand.

Clarity: Scenarios written in Gherkin (the language Cucumber uses) provide clear and concise descriptions of desired behavior, making it easier to understand the purpose of each test.

Reusability: Step definitions in Cucumber can be reused across different scenarios, reducing duplication and improving maintainability.

Automation: Cucumber tests can be automated, allowing for continuous integration and regression testing.

Living Documentation: Scenarios serve as living documentation, providing a clear picture of the system's behavior over time.

Focus on User's Perspective: BDD emphasizes writing tests from the user's perspective, ensuring that the software meets user expectations.

Disadvantages of BDD with Cucumber:

Complexity: Implementing BDD with Cucumber can introduce complexity, especially in large projects or teams unfamiliar with the approach.

Overhead: Setting up Cucumber and writing scenarios in Gherkin can be time-consuming, especially for small projects where the benefits may not justify the effort.

Maintenance: As with any test automation framework, maintaining Cucumber tests requires effort, especially when the application undergoes frequent changes.

Learning Curve: Team members unfamiliar with BDD or Gherkin may require time to learn the principles and practices, potentially slowing down initial adoption.

Misuse: Cucumber can be misused, leading to scenarios that

Version Control

Q: Why the version control tools used? Give examples for version control tools?

Ans: In software engineering, version control (also known as revision control, source control, or source code management) is a class of systems responsible for managing changes to computer programs, documents, large web sites, or other collections of information. Version control is a component of software configuration management.

Ex: gitHub, bit Bucket, VSS, SNV, Perforce, TFS etc

Q: How to install and configure the gitHub & client?

We need to register to gitHub. Go to https://github.com/ and register

-> After registration create a separate space for youself which is k.a., Repository.

-> We need to access the gitHub from our local PC, hence we need to install git client as well.

go to https://git-scm.com/download/ website--> select the OS & bits--> Download the exe file and

install

Q: What is repo?

Ans: It is a place in the version control tool server allocated you to use it for your project.

A repository contains all of your project's files and each file's revision history. You can

Discuss and manage your project's work within the repository.

Q: How to create the repo?

Ans: Go to gitHub-->Login-->click on New button --> Give name to the repository-->Click on

Create Repo button

Q: How to add the initial framework to the git Repo?

Ans:

Create a working framework & follow the below commnads to add the framework to the git repository.

open cmd-->navigate to your framework workspace-> execute below commands:

git config --global user.name <your name>

git config --global user.email <your email>

->Create a local repo.

git init

go to git folder->rightclick select open git bash here-> coomnad prompt will open

->add your framework to the git staging as follows:

git add *

-> commit your framework

git commit -m "Initial framework draft"

-> connect to the remote remote repo in git

git remote add origin <git repo url>

Ex: git remote add origin <https://github.com/gopal2602/MorningBatch.git>

-> push your framework to the git repo

git push -u origin master

git switch -c main master

git push -f origin main

Q: How to clone the repo?

Ans: Clone means downloading the framework from the git repo into your local PC. For this you need:

1. git repo URL.

2. The client must have installed

Go to the location where you want to download the framework. In the selected folder, right click

and select git bash here. It will opens the git bash cmd.

git clone <repo URL> --branch <branch name>

Q: How to creat the local branch in the git?

Ans: By default git will have master/Main branch. Whenever we clone the master/Main branch,

we must create our local branch. Bcoz we must not directly work on the master/Main branch.

go to your local copy--> go to project--> right click and select git bash here.

It will open the git bash window & shows that you are in branch:

create your local branch as follows:

git checkout -b <localBranchName>;

This above command will create the new local branch and switch to it.

Q: How to switch from one branch to another branch?

Ans: we are in master branch. I want to switch to ggudi branch. Then we need to run below commmand:

git checkout <Branch Name to switch>

Q: Why we have to create the local branch? Can't we directly use the master/Main branch?

Ans: No, we can't do changes directly in the master branch. Because master branch is the master copy.

So we have to create a local branch, do changes, commit and push them back to local branch.

Then create a PR (Pull Request) to merge your changes to master branch.

Q: How to stash the changes?

ans: git stash

Q: How to discard the changes?

Ans: git restore <file path>

Q: How to add the changes?

Ans:

git add <file path>

Q: How to add, commit and push our changes to the local branch?

Ans: git status: it will show all the list of files which got modified

case: 1=> Using git Bash command prompt window:

1. git add <file path>
2. git commit -m <message>
3. git push --set-upstream origin <branchName>

from second time onwards you have to use: git push

case: 2=> Using intellij tool

1. do the changes.
2. go to git menu-->Commit. It will show all your changes.
3. select /deselect the files based on your changes
4. Provide appropriate comments
5. click on "Commit and Push" button

Q: How to merge my local branch changes to master branch?

Ans: 1. first push your changes to your local branch

2. Create a PR (Pull Request) & assign it to the reviewer

3. The reviewer will review your code & approve the changes

4. Once approved, merge with master

Q: How to take the latest code from master/any branch?

Ans: first login to git bash here--> switch to the branch from where you want to take the update.

& pull from there.

Note: Currently i'm in local branch. I have to take the changes from master branch

git checkout master

git pull

git checkout <your branch>

git rebase master

Note: If you want to pull the master changes from Intellj, do following steps:

Go to git menu-->pull-->provide appropriate branch name from where you want to pull the latest

changes.-->Click on pull button

Q: How to compare my local changes with Head version (Server copy)?

Ans:

go to eclipse, select the file which have modified (it shows > symbol)--> right click -->

Compare with--> Head Revision

go to intellij, select the file/whole framework which you have modified--
>

right click-->git-->Compare With Branch...-->select origin/main-->

It will show all the files which got modified. Open one by one and we can see the changes.

Q: How to replace the local changes with HEAD version?

Ans:

go to eclipse, select the file which have modified (it shows > symbol)--> right click -->

Replace with--> Head Revision

CI/CD: Contineous Integration & Contineous Development

CI tool advantages:

1. help streamline the build process with the use of automatic triggers from the version control system

2. to find and address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates.

3. enables better transparency and insight into the process of software development and delivery

Q: How CI/CD pipeline works:

Dev writes the code--> push the code to GitHUB--> Then pipeline will start:

once dev push the code--> CI/CD tool will trigger, compile your code--> Create a build-->Deploy the build in QA and Dev environemnt and trigger the automation framework-->Monitor the framework execution-->creates a report and sendout mail to all the stakeholders.

Our case:

Write my framework--->put framework in gitHUB-->Configure Jenkins to manage our test automation process--> Jenkins will compile and run our framework when ever we do code change-->It will monitor the execution and finally prepare the reports and send a mail to all.

We will discuss Jenkins as a CI tools.

Q: How to install the jenkins?

Ans: It can be installed in 2 ways:

(a) using .exe file

(b) .war file

Deploy jenkins using .war file:

1. Go to goole and search for "Download jenkins .war file"

2. Click and open the official link https://updates.jenkins.io/download/war/

3. Download the .war file by clicking on suitable version

4. Place the .war file in an empty folder

5. Open the command prompt and navigate to the jenkins folder path.

6. Execute the below command to deploy the .war file

java -jar <jenkins .war file name> --httpPort=<portNumber>

wait for the message "Jenkins is fully up and running"

7. Open browser and type the url as "http://localhost:<portNumber>/".

8. Once Jenkins opens it will ask for the Admin password. Copy the admin password from below location and provide to the jenkins and click on continue button:

C:\Users\<ComputerName>\.jenkins\secrets\initialAdminPassword

8. Click on "Install Suggested Plugins" block. Wait for the plugins to install

9. Once default plugins are installed, jenkins will ask you to register & create a Admin user credentials.

10. Configure java, maven in jenkins by navigating to Dashboard-->manage Genkins-->Tools.

11. Install the maven integration plugins by navigating to Dashboard-->manage Genkins-->Plugins.

click on "Available Plugins"-->Type 'Maven Integration'-->Select the checkbox of 'Maven Integration' and click on "Install" button.

12. Create a maven job and configure gitHub:

Go to Dashboard-->click on 'New Item'-->Give a suitable name to the jenkins job-->Select 'Maven Project'-->Click on "OK button.

Configure below things:

(a) Give suitable description

(b) Source Code Management: select 'Git' radio button.

Provide the git repo url.

(c) Build Triggers: select build periodically checkbox.

ex: * * * * * every minute

H/10 * * * * every 10 minutes

H/15 * * * * every 10 minutes

(d) Build:

poll SCM checkbox

click on Apply and Save buttons

Test Automation Framework(TAF):

It is an systematic approach of converting manual test cases into automation scripts with the help of tools & execute them contineously without any manual involvement is k.a., TAF

Characteristics of a Good Framework:

1. Reusability
2. Scalability
3. Effective reporting mechanism.

4. Cross Platform OR cross browser support

5. Reduce execution time

6. Easy to use OR user friendly

7. Easy to maintain the framework

Types of Framework:

1. Linear frameworks

2. Datadriven frameworks

3. Keyword driven framework

4. Hybrid framework

5. Modular Framework

6. Cucumber Framework (BDD - Behaviour Driven Development)

7. POM (Page Object Model)

Folder Structure:

D:\SG_Maven_Framework

|

V

SG_Mvn_Project

|

V

Inside the main

src

1. pages

Inside test resources

2. config.properties

Inside test src

1. driver
    (a) DriverScript.java/TestRunnerClass.java
2. common
    (a) AppIndependentMethods.java
    (b) AppDependentMethods.java
    (c) Datatable.java
3. baseclass
    (a) UserModuleMethods.java
    (b) TaskModuleMethods.java
4. stepdefinition
    (a) UserModuleStepDefinition.java
    (b) TaskModuleStepDefinition.java

5. reports
    (a) ReportUtils.java

Inside test resources

featurefile
    (a) userModule.feature
    (b) taskModule.feature

folders structure

TestData->It contains Excel sheets corresponding to the module names mentioned in the ExecutionController.xlsx. Here test data is module specific.

Configuration-> Config.properties file. It is used for reading the global/Master data. (The data which is used across the framework)

Results

-ScreenShots: Upon failure the screenshot will be captured & placed inside this folder.

-HTMLReport: HTML report (Extent Report) for the overall execution status

Framework Explanation:

Our framework is a hybrid framework which is a combination of :

-Data driven framework

-Keyword driven framework

-Modularization framework

-BDD cucumber

Data driven: All the inputs viz., Test data, Object locators, Configuration etc are reading from the external file systems like excel, .properties file etc.

Keyword driven: All the use level & application level actions are converted into keywords (methods) & these keywords are used/arranged to achieve the automation.

Modularization: This approach helps to achieve reusability. Here the application is devided into modules, modules are devided into methods etc. This helps to make framework independent.

BDD cucumber: Cucumber is a software tool that supports behavior-driven development. Central to the Cucumber BDD approach is its ordinary language parser called Gherkin. It allows expected software behaviors to be specified in a logical language that customers can understand.

Our framework is a Maven framework. Hence execution will starts from pom.xml file.

It can also triggered from jenkins (CI) tool.

pom.xml (Project Object Model) does following:

1. It will downloads the dependencies

2. Compile the source code

3. Builds the project

4. Set the build path for the jars located in local .m2 repository

5. Then gives the control to testng.xml file

Once testng.xml is invoked, It will run the CucumberTestRunner.java file based on the cucumber tags specified.

CucumberTestRunner.java class file will have following annotations:

@BeforeSuite: The method with this annotation will load all the class files OR instantiate object to all the class files in our framework.

All the cucumber feature files will have their corresponding stepDefinition class files and every stepdefinition class files will have its corresponding BaseClsss files (BaseClsss files will have module specific reusable methods)

As the framework contains Modularization approach, the methods, testdata & testscripts are module specific. So when we execute the testscripts it will go to its corresponding module consumes all the

methods, test data, object locators, utility metohds etc to execute. After the execution of the test scripts it will proides 2 types of results:

HTMLReports:

2 types of reports:

(a) extent report: Detailed step by step reports with proper screenshots whereever applicable and where the step fails

(b) cucumbe report: is a high level report corresponding to the Gherkin steps specified in feature file

Q: How do you analyse the failures?

Ans: First we will re-run the failed tests. After re-run most of the false failures will be eliminated. Then we will debug the actual failues with the help of detailed results and screenshots available in the results.

Q: How to rerun the failed one in testNG framwork?

Ans: As we are using unti testing tool testng, it will create a separate "testng-failed.xml" file under /test-output folder for all the failed tests. We need to run this "testng-failed.xml" file in order to re-run only the failed tests.

Q: Can we achieve parallel execution using testng?

Ans: Yes. We can run multiple testng scrips together based on the thread-count values.

Q: Tell me about yourself?

Ans:

Hi, My name is <Your Name>, I am from <Native place>. I have completed my education <DegreeName> in <college Name> in the year <year of pass>.

Immediately after completion of degree I got an apportunity work in <company Name> as a software Test Engineer.

Currently I have completed around <years of exp> years in <company Name>. During this periods I have completed around <numbers> projects.

My current project is <Current project Name>. Here my roles and Responsibilities are:

1. I have involved in design & development of hybrid framework using selenium web driver & core java

2. Involved in Enhancing my framework by implementing BDD (Cucumber) which helps to provide readability to even non-tech people about the framework.

3. Involded in Design & Development of mobile automation framework using Appium for both Android & iOS

4. Involved in Webservice automation using Postman (Manual) & restAssured + Java

5. Involved in automation of DB using JDBC connection

6. Helping the team members as well as cross team members to learn & ramp up on the frameworks & programming stuffs by providing trainings OR KT.

7. Involved in preparation of automation POC for the new customer.

8. Understanding customer requirements, writing test plan & test cases by applying test cases design techniques.

9. With the help of automation we have reduced around 50 % of human efforts.