# What are the different API available in selenium?

NOTE: API can be in the form of interface, classes, methods and commands.

Navigation

Delete cookies

Resizing the window

Searching the web element

Window handling

Alert handling

Iframe handling

Mouse handling

Keyboard handling

Wait

Dropdowns

Locator

What are the verification points available in selenium?

There is no count fix of verification points in selenium, when we are comparing between two things to verify its works expected or actual vs expected it could be Asserts and those method, which are returning Boolean values true or false.

Methods available in selenium, which are returns Boolean values i.e. true or false.

Is Displayed ();

Is Selected ();

Is Enabled ();

# What is selenium web driver?

Selenium web driver is user friendly API which are used to automate the web application to verify that it works as excepted, its support multiple browser, multiple Operating systems and multiple languages but it doesn’t support window-based application, so selenium web driver API or interface has multiple methods

1. driver.get(“”);
2. driver.close();
3. driver.quit();
4. driver.findelement();
5. driver.findelements();
6. driver.getTitle();
7. driver.getWindowHandle();
8. driver.getWindowHandles();
9. driver.getCurrentUrl();

What does this code?

Web driver driver=new Chrome Driver (); //We are creating the object of Chrome Driver class.

Chrome driver is the class which implementing to web driver interface, so web driver is the parent of all these classes. (Dynamic Poly.)

What is the super interface of web driver interface?

Search context.

Why should write like this Web driver driver=new Chrome Driver (); instead of Chrome driver driver=new chrome driver (); because if use this we must write repeatedly this above line to launch every browser.

What is automation testing?

Automation testing is the process of testing the software by using automation tool to find the defects, executing the test script and generating the result are performed automatically by automation tool, which can be QTP/UFT, selenium web driver etc.

What are the advantages of automation testing/ Why did you select Selenium WebDriver for front end automation of your application?

1.Selenium web driver is open source and totally free of cost.

2.Selenium support multiple languages like java, python, java script, Perl and C#

3.Selenium support multiple operating system to check the compatibility of application

4.Selenium supports multiple browsers to perform the cross-browser testing.

5. Using selenium can be handle the multiple scenarios like mouse and keyboard handling.

6.In selenium we can implement the third-party tool like TestNG to generate the reports and Appium for mobile testing.

7.we can save the time, resources and money as well.

8.It takes less maintenance and more reliable.

2. What are different selenium components?

* IDE – It’s a play and record tool and distributed by Firefox
* RC– Selenium RC is a server that allows to user to create test scripts in a desired programming language, RC communicated with browsers through server.
* Web driver–Selenium Web driver has a various advantage over RC, Web driver directly communicated with browser.
* Grid – In Grid we can execute the test script on multiple operating system and browsers through remotely.

3. What are testing type that can be supported by selenium?

* Functional :

Regression, sanity, smoke and end-to-end.

1. 4. What are limitations of selenium/ What are the limitations of Selenium WebDriver?

* Selenium cannot test mobile based applications.
* Selenium cannot test window-based applications.
* Selenium cannot test Captcha and Bar code reader.
* Selenium cannot generate the test reports.
* In selenium there is no ready vendor support.
* User is expected to possess prior to programming language.

How you will setup a Selenium WebDriver project from scratch?

* **Java** as the programming language
* **TestNG** as the assertion framework
* **Maven** as the build tool
* **WebDriver** as the browser automation tool
* **IntelliJ** as the IDE

5. What are the different types of locators in selenium?

* ID
* Name
* Link text
* Css Selector
* Xpath
* Class Name
* Tag Name
* Partial Link text

6. How to launch the browser in selenium with syntax?

* Chrome

System.set.property (“webdriver.chrome.driver”,”Path of .exe browser”)

Webdriver driver = new ChromeDriver();

* FF

System.set.property (“webdriver.firefox.driver”,”Path of .exe browser”)

System.*setProperty*("webdriver.gecko.marionette"," Path of .exe browser");

Webdriver driver = new FireFoxDriver();

* IE

System.set.property (“webdriver.ie.driver”,”Path of .exe browser”)

Webdriver driver = new InternetExplorer();

7. How to maximize the browsers window in selenium?

* Driver. manage().window().maximize();

8. How to open the Web application URL in selenium?

* Driver.get(“Url of application”);

9. What is Xpath and How many types of Xpath with syntax?

Xpath is a locator which is used to identify the web element based on html

tag, attribute & text.

1. Single / X path OR Absolute - Absolute Xpath is located the web element from root node to required child node. In real time application we should not use absolute Xpath because it is use index.

Example- html/body/div[1]/section/div[1]/div/div/div/div[1]/div/div/div/div/div[3]/div[1]/div/h4[1]/b

1. Double // X path OR Relative –Relative X path starts from the middle of the HTML DOM structure. It starts with the double forward slash (//), which means it can search the element anywhere at the webpage

**Syntax-: //tag name [@attribute=’attribute value’]**

10. How many types of x path function?

(i)contains ()

Syntax=//htmltag[contains(attribute(),'attributevalue')]

Xpath=//\*[contains(text(),'guru99.com')]

(ii)text()

Syntax:-//htmltag[text()=’textvalue’]

Xpath=//input[text()=’abc’]

(iii) starts-with()

Syntax:-//htmltag[starts-with(@attribute,’attributevalue’)]  
Xpath=//label[starts-with(@id,'message')]

(iv) following-siblings

Syntax:-//tagname[@attribute-‘attributevalue’]/following-sibling::tagname[@attribute=’attributevalue’]

(v) preceding-siblings

Syntax:- //tagname[@attribute-‘attributevalue’]/preceeding -sibling::tagname[@attribute=’attributevalue’]

(vi) descendent-function

Syntax:-//tagname[attribute=’attributevalue’]/..//tagname[@attribute=’attributevalue’]

12. How to handle multiple windows and switch to windows?

13. How to perform mouse handling and types of methods?

\*By actions class we can perform mouse handling and some keyboard action as well.

\*Methods -:

\*click (target webelement)

\*clickandhold(target webelement)

\* movetoelement(target webelement)

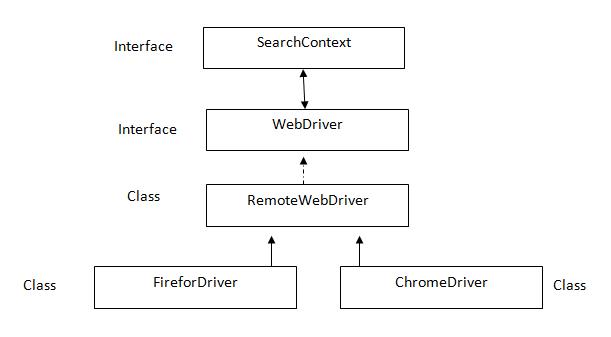
\*release

\*Drag and drop (source target, destination target)

\*drag and drop (sourcetarget,”x”,”y”)

\* key up and key down

NOTE -: We have to use build.Perform to invoke the actions class because in actually build.perform performing the action.

What is the structure of web driver API?

**SearchContext** is super most interfaces which are extended by **WebDriver** interface. Abstract methodsof these two interfaces is implemented in **Remote WebDriver** class and overridden in respective browser classes such as FirefoxDriver, ChromeDriver, InternetExplorerDriver, SafariDriver etc.

How to find the location of web element in a web page?

Ans :- WebElement wb=driver.findElement(By.id(“abc”));

Point p=wb.getlocation();

Int x=p.getX();

Int y=p.gety();

System.out.println(x+””+y);

Difference between find element and find elements?

Ans :-(i)Find element handles single web element and Find elements handle multiple element.

(ii)Find element return type is web element and find elements return type of List<web element>.

(iii)If findelement() fails to find the web element.It gives “No such element found exception” error. And if findelements() fails to find the webelement,then it gives “Empty list”.

How to set the position of browser ?

For browser:-

Point p=new Point(x,y);

Driver.manage().window().setPosition(p);

How to resize the browser?

Dimension d=**new** Dimension(200, 200);

Driver.manage().window().setSize(d);

14. What is TestNG and advantages and disadvantages?

TestNG is advance or open source or TDD framework which is inspired by Junit and Nunit and TestNG framework developed for developer to write the Junit test cases but TestNG introducing more functionality they will give more features rather than to Junit and Nunit and That make it more powerful and easier to use.

Advantages-:

1. They will provide html reports so you can use and see the reports at any time.
2. They will provide more annotations so you can use annotations based on requirement multiple annotations-:

* @before suite
* @after suite
* @before test
* @after test
* @before class
* @after class
* @before method
* @after method
* @test
* @data provider

1. Provide priority/ sequence – We have one keyword priority in TestNG with the help of this we can execute our test cases @Test by priority keyword and priority should be define at test case level @Test Ex: @Test (priorities=2) and smallest no priority will be execute first even negative value given
2. Dependency – We have one keyword ‘depend On Methods’ in TestNG with the help of this one test cases is dependent on other test cases and ‘depend On Method’ should be define at test case level @Test Ex: @Test (dependOnMethod=”LoginPage”), here you have to put that test case of method name, so if login page is got passed then dependent test cases will be pass either will be skipped.
3. Grouping – We have one keyword groups in TestNG with the help of this we can define test cases grouping by groups keyword and groups should be define

at test case level @Test Ex: @Test (groups=” Test”) Like: Title test cases, logo test cases, functional test cases and GUI test cases etc.

1. Data provider
2. @Parameters– This feature is a part of data driven, when you need the test data or configuration or env. data from other resources or file we can use @parameters annotation in TestNG and in the form of parameter name and value in .xml file.

Ex:

* **Public class** Parameterization {

WebDriver driver;

@Test

@Parameters({"Url","Username","Password"})

**Public void** f(String Url,String Username,String Password) {

System.*setProperty*("webdriver.chrome.driver", "D:\\Old\_Laptop\_Back\_Up\\Tools\\chromedriver.exe");

driver=**new**ChromeDriver();

driver.manage().window().maximize();

driver.get(Url);

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(30,TimeUnit.***SECONDS***);

driver.findElement(By.*id*("email")).sendKeys(Username);

driver.findElement(By.*id*("pass")).sendKeys(Password);

driver.findElement(By.*id*("u\_0\_2")).click();

* <?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!DOCTYPEsuiteSYSTEM"http://testng.org/testng-1.0.dtd">

<suitename=*"ParameterSuite"*>

<testname=*"DataParameterTest"*>

<parametername=*"Url"*value=*"https://www.facebook.com/"*></parameter>

<parametername=*"Username"*value=*"9866740237"*></parameter>

<parametername=*"Password"*value=*"12345"*></parameter>

<classes>

<classname=*"Parameters.Parameterization"*>

</class>

</classes>

</test>

* </suite>

1. Cross browser testing –

* WebDriver driver;

@Test

@Parameters({"browserName"})

**Public void** f(String browserName) {

**if**(browserName.equalsIgnoreCase("chrome"))

{

System.*setProperty*("webdriver.chrome.driver", "D:\\Old\_Laptop\_Back\_Up\\Tools\\chromedriver.exe");

driver= **new**ChromeDriver();

driver.manage().window().maximize();

driver.get("http://www.google.com");

driver.manage().timeouts().implicitlyWait(30, TimeUnit.***SECONDS***);

}

**elseif**(browserName.equalsIgnoreCase("Firefox"))

{

System.*setProperty*("webdriver.gecko.driver", "C:\\Users\\aa\\Downloads\\geckodriver-v0.19.1-arm7hf\\geckodriver");

driver= **new**FirefoxDriver();

driver.manage().window().maximize();

driver.get("http://www.google.com");

driver.manage().timeouts().implicitlyWait(30, TimeUnit.***SECONDS***);

}

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

* <?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<suitename=*"Testing"*>

<testname=*"ChromebrowserTesting"*>

<parametername=*"browser"*value=*"chrome"*>

<classes>

<classname=*"CrossbrowserTesting.CrossbrowserTesting"*>

</class>

</classes>

</parameter>

</test>

<testname=*"FFbrowserTesting"*>

<parametername=*"browser"*value=*"Firefox"*>

</parameter>

<classes>

<classname=*"CrossbrowserTesting.CrossbrowserTesting"*>

</class>

</classes>

</test>

</suite>

1. Logs
2. Expected exception- We have one keyword expected exception with the help of this we can ignore exception by Web driver should be define at test case level @Test Ex: @Test (expected exception=No such element. class)

How do you run a test method multiple times?

OR

How to execute a single script multiple times?

Using “InvocationCount”.

Ex:- class Demo

{

@Test (invocationCount=4)

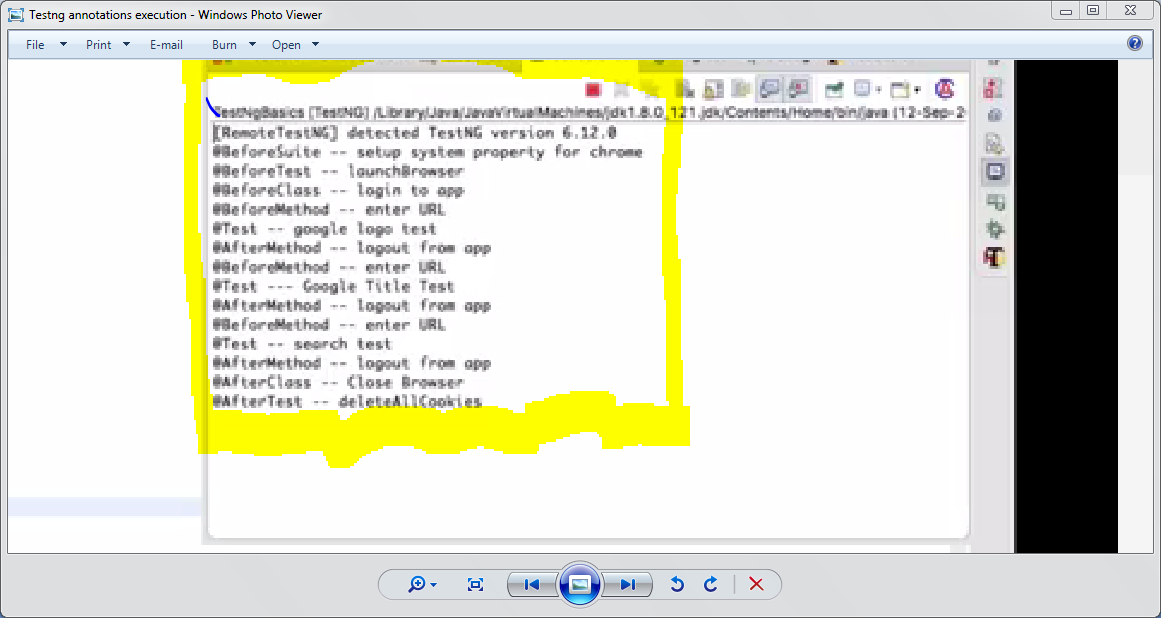
Public void test()

{  
System.out.println (“Hello test”);

}

}

15. In which order or sequence all TestNG annotation will be run in TestNG?



16. What is TestNG .xml file and use in selenium?

TestNG framework provided one good feature that is TestNG .xml file in .xml file you can configure our suite name, test name, packages name, classes name and method name with help of .xml file we can run the all classes which is created in packages in one shot.

Ex.

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!DOCTYPEsuiteSYSTEM"http://testng.org/testng-1.0.dtd">

<suitename=*"ParameterSuite"*>

<testname=*"DataParameterTest"*>

<classes>

<classname=*"AlertPopup.AlertPopup"*>

</class>

<classname=*"WindowPopup.WindowPopup"*>

</class>

</classes>

</test>

</suite>

NOTE- We can execute .xml file from simple java project and maven project,

No need to add additional things to run from maven.

17. How to run the .xml file from bat file in simple java and maven project?

\*Create one folder in project location then put jar file in that folder

Selenium.jar and testing.jar

\*Open notepad and put this below written code

set projectLocation=F:\Selenium\TestNGBatchExample cd %projectLocation% set classpath=%projectLocation%\bin;%projectLocation%\lib\\* java org.testng.TestNG%projectLocation%\testng.xml pause

which is written in above code set project location you have to put your project location after that in the last line you have to put your .XML file name. Which is marked by yellow colour.

18. How to handle alert popup in selenium with syntax and types of alert?

* //Simple alert

Alert SimpleAlert= driver.switchTo().alert();//Switch to alert

System.***out***.println("Simple Alert text is => "+SimpleAlert.getText());// Displaying alert msg.

SimpleAlert.accept();//Accepting alert

* //Confirmation alert

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

System.***out***.println("Confirmation Alert is => "+ConfirmationAlert.getText());

ConfirmationAlert.dismiss();

* //Prompt Alert

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

System.***out***.println("Prompt Alert text is => "+PromptAlert.getText());

PromptAlert.sendKeys("Sure Y Not");

PromptAlert.accept();

19. What is API?

\*API is a collection of routines, tools, protocols that together are required for building the software application. Any system software or application software, which consists of multiple APIs, can perform Application Programming Interface (API) testing.

20. Why API needed AND API Testing?

\*In API we can create or exchange the data between twoapplications by using API or we can say indirectly we are hitting or access to business logics or methods of application to generate the output without any access of source code

* API means it always evaluates used when two system interacting with each other in that case we use API and these two system totally independent to each other means(technology wise ) but they want to use extra features provided by companies and system. In that case we use API.API medium is JSON /XML file.

21. What are Web services?

In web services we can create and exchange the data or communicate between two applications through XML or JSON because both are light weighted languageand may have both applications made on different- different technology, so we can access or communicate with each other over the HTTP.

See some online portal they are providing all airlines availability means how many seats are available, fare and date etc.Online portal are not requesting to source code because Airlines Companywon’t provide the source code they can give only API so we can send the request and take the response from application of method or functionality through API.

* When API is getting call over the network or over the http call. It becomes web services. Everything through web services happening over network or HTTP call.
* Where we are not providing any jar files. And we are communicating over network or HTTP call. This type of API called web services.

What are the HTTP methods?

GET, POST, PUT, DELETE these are the HTTP methods.

HOW WILL YOU ACCESS API?

WE CAN ACCESS BY API UI AND THIRD PARTY TOOL POSTMAN, SOAP UI, JMETER, BROWSER ETC.

What is CRUD?

C - CREATE - POST –

POST CALL WILL BE USE TO NEW ENTITY

IN POST CALL YOU HAVE TO PASS URI, PAYLOAD AND PARAMETER (HEADERS)

R – RETRIVE – GET –

GET CALL WILL BE TO RETRIVE OR FETCH THE ENTITY

IN GET CALL YOU NEDD TO PASS ONLY URI OR PATH PARAMETER OR QUERY PARAMETER

U – UPDATE – PUT –

GET CALL WILL BE USE UPDATE TO ENTITY

IN PUT CALL YOU NEED TO PASS URI, PAYLOAD AND PARAMETER (HEADERS)

D – DELETE – DELETE

WHAT IS URI?

URI IS COMBINATION OF URL+PATH PARAMETER+QUERY PARAMETER

PATH PARAMETER – //HTTP://API.COM/SERVICE/ACCOUNT/1

IN ABOVE URI WHICH IS MARKED BY COLOUR /1MEANS YOU WANT TO RETRIVE THE SPECIFIC ENTITY.

QUERY PARAMETER – //HTTP://API.COM/SERVICE/ACCOUNT/?PAGE=2

IN ABOVE URI WHICH IS MARKED BY COLOUR /? PAGE=2MEANS YOU WANT TO FILTER OUT.

What should we pass in API through HTTP? / WHAT YOU NEEDED FOR SEND THE API REQUEST?

1. URI
2. HEADERS
3. PAYLOAD – IN WHICH FORM YOU WANT – JSON/XML

WHAT WILL YOU GET IN API RESPONSE?

1. STATUS
2. RESPONSE PAYLOAD
3. STRING MESSAGE

WHAT IS THE FORM OF PARAMETER OR HEADERS?

KEY AND VALUE AND THAT SHOULD BE PASSES IN THE FORM OF STRING

WHAT IS THE STATUS CODE?

WHENEVER YOU REQUESTED TO API THEN YOU WILL GET STATUS RESPONSE

1. 200 - OK – MEANS YOU ARE HITTING CORRECT API OR SUCCUSSFULLY

GENRALLY 200 OK STATUS YOU WILL GET IN GET CALL

1. 201 - CREATED – MEANS YOU CREATED NEW ENTITY SUCCUSSFULLY

GENRALLY 201 CREATED STATUS YOU WILL GET IN POST CALL

1. 204 – NO CONTENT – MEANS YOU WERE DELETED OR NOT AVILABLE IN SERVER

GENRALLY 204 YOU WILL GET IN DELETE CALL

1. 400 – BAD REQUEST – MEANS YOU ARE PASSING WRONG END POINT

URI AND NOT VALID FORMAT ETC.

1. 404 – NOT FOUND - MEANS YOU ARE CALLING TO THAT API THAT IS NOT

AVAILABLE IN THE SERVER

GENRALLY 404 NOT FOUND STATUS YOU WILL GET IN GET, DELETE AND PUT CALL

1. 500 – SERVER INTERNAL ERROR – MEANS YOU ARE HITTING API BUT SOME INTERNAL

ERROR

1. 401 – AUTHENTICAION ERROR

NOTE – EVERY SERVER HAS CONNECTED WITH DATABASE

WHAT IS REST API

WHY API IS GETTING POPULAR?

22. What is assert and how many types of Asserts in selenium with syntax?

Assertion is used to perform various kinds of validation and its help us to decide weather

the test case has passed or failed.

* Hard assertion – In hard assert if execution is getting fails or throws exception then rest of the test script will not execute.
* Syntax -:

Assert.assertEquals(exc, act, msg);

* Soft assertion - In soft assert if execution is getting fails or throws exception then rest of the test script will be execute in any how condition.
* Syntax -:

**public** **class** Assert {

WebDriver driver;

@Test

**public** **void** test1() {

System.*setProperty*("webdriver.chrome.driver","D:\\Tools\\chromedriver\_win32\\chromedriver.exe");

driver=**new** ChromeDriver();

driver.get("http://www.google.com");

SoftAssert softassert1=**new** SoftAssert();

String title=driver.getTitle();

softassert1.assertEquals(title, "Doodle", "invalid title");

}

@Test

**public** **void** test2() {

SoftAssert softassert2=**new** SoftAssert();

String button=driver.findElement(By.*name*("btnK"));

button.getText();

softassert2.assertEquals(button, "Google Search", "wrong name");

softassert2.assertAll();

}

}

NOTE – 1. AssertAll (); collect all the test cases result weather passed or failed in object of SoftAssert class and if fails throw exception

2. assertAll (); we have to write reference variable.assertAll(); at end of the script if you have single test cases and if you are putting reference variable.assertAll(); in mid of the script then if script result getting passed then will execute rest of the script if script result getting failed rest of the script will not execute.

3.assertAll(); if we have more than one test cases then for each and every test cases just create reference variable of SoftAssert class and end of the test cases just write reference variable. assertAll ();

4. If you are not putting reference variable. assertAll(); then test cases will be marked as passed even though test cases are getting failed.

23. Difference between Assert and Verify?

* In assertion if execution is getting fails or throws exception then rest of the test script will not execute.
* In verify even though if execution is getting fails or throws exception then rest of the test script will be execute in any how condition.

24. How to verify error message in selenium?

driver.findElement(By.*xpath*("//span[text()='Next']")).click();

String ActualErrorMsg= driver.findElement(By.*xpath*("//div[contains(text(),'Enter an email or phone number')]")).getText();

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

String ExpectedErrorMsg="Enter an email ";

SoftAssert assertion =**new** SoftAssert();

assertion.assertEquals(ActualErrorMsg, ExpectedErrorMsg,"Invalid Error Msg ");

assertion.assertAll();//we have to use end of the program further rest of the script will not run.

25. How many types of navigation command in selenium?

* Driver.navigate.to(“Url”)
* Driver.navigate.forward();
* Driver.navigate.back();
* Driver.navigate.refresh();

26. What is the difference between driver.get (); and driver.navigate.to ()?

The difference is driver.get method will wait until page has full loaded but driver.naviate.to method will not wait until page has fully loaded or not and navigation API has multiple method for browser window which is in below but driver.get has no method.

1.forward

2.back

3.refresh

4.to

**public** **class** Navigation {

WebDriver driver;

@Test

**public** **void** f() **throws** InterruptedException {

System.*setProperty*("webdriver.chrome.driver","D:\\Tools\\chromedriver\_win32\\chromedriver.exe");

driver=**new** ChromeDriver();

//driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(20, TimeUnit.***SECONDS***);

driver.get("http://www.half.ebay.com");

driver.navigate().to("http://www.google.com");

Thread.*sleep*(2000);

driver.navigate().back();

Thread.*sleep*(2000);

driver.navigate().forward();

Thread.*sleep*(2000);

driver.navigate().refresh();

}

}

27. How to verify title in selenium?

We can verify by -:

* driver.getTitle ();

28. How will you find the web element is available or not on web application?

We can find by -:

* driver.isDisplayed (); -This method is used to element is present or not on the webpage.
* driver.isEnabled (); -This method is used to element is enabled or not on the webpage.
* driver.isSelected (); - This method is applicable for checkbox, radio button and dropdown is selected or not on the webpage.

**public** **class** ElementPresent {

WebDriver driver;

@Test

**public** **void** f() {

System.*setProperty*("webdriver.chrome.driver","D:\\Tools\\chromedriver\_win32\\chromedriver.exe");

driver=**new** ChromeDriver();

driver.get("https://www.freecrm.com/register/");

((JavascriptExecutor)driver).executeScript("scroll(0,200)");

WebDriverWait wait= **new** WebDriverWait(driver,6 );

WebElement displayed=wait.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("//button[text()='SUBMIT']")));

**boolean** statusdisplayed =displayed.isDisplayed();

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

WebDriverWait wait1= **new** WebDriverWait(driver,6 );

WebElement enabled=wait1.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("//button[text()='SUBMIT']")));

**boolean** enabledstatusfalse =enabled.isEnabled();

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

WebDriverWait wait2= **new** WebDriverWait(driver,6 );

WebElement agreeTerms=wait2.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("//input[@name='agreeTerms']")));

agreeTerms.click();

WebElement enabled1=wait2.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("//button[text()='SUBMIT']")));

**boolean** enabledstatustrue =enabled1.isEnabled();

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

WebElement agreeTerms1=wait2.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("//input[@name='agreeTerms']")));

agreeTerms1.click();

WebElement selected=wait2.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("//input[@name='agreeTerms']")));

**boolean** selectedstatusfalse =selected.isSelected();

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

WebElement agreeTerms2=wait2.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("//input[@name='agreeTerms']")));

agreeTerms2.click();

WebElement selected1=wait2.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("//input[@name='agreeTerms']")));

**boolean** selectedstatustrue =selected1.isSelected();

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

}

}

30. What are challenges you have faced in selenium?

* In selenium there is no ready vendor support we have to do R&D on internet

But in commercial tool this facility is there.

* We need to learn one prior programming Language java, python php etc working for

Selenium.

* Difficult to identify dynamic objects.
* Selenium is not given any interface or class for reporting we have to dependent on third party tool.
* Time synchronization problem when will use implicitly, explicitly and fluent wait

Because this thing is effected on your application or performance.

* Unexpected error when launching the IE browser or not able to identify the web elements.
* No dependency between selenium version and browser & browser driver versions

Sometimes browser is not getting launch. Or gives unexpected error due to no dependency.

* Ajax component problem because selenium not waiting to load the Ajax images

31. How to handle page scroll bar in selenium?

We can handle by JavaScript Executor interface

* **public** **class** Page\_Scroll\_bar {

WebDriver driver;

@Test

**public** **void** f() {

System.*setProperty*("webdriver.chrome.driver","D:\\Tools\\chromedriver\_win32\\chromedriver.exe");

driver=**new** ChromeDriver();

// System.setProperty("webdriver.gecko.driver","D:\\Tools\\geckodriver-v0.16.1-win64\\geckodriver.exe");

//driver=new FirefoxDriver();

driver.get("http://docs.seleniumhq.org/download/maven.jsp");

((JavascriptExecutor)driver).executeScript("scroll(0,500)");

driver.findElement(By.*id*("close")).click();

}

}

32. How to retrieve attribute value means (id, name, link text ,type etc.) or text from textbox and capture to element text or label text?

**public** **class** GetAttribute\_GetText {

WebDriver driver;

@Test

**public** **void** f() {

System.*setProperty*("webdriver.chrome.driver","D:\\Tools\\chromedriver\_win32\\chromedriver.exe");

driver=**new** ChromeDriver();

driver.get("http://toolsqa.com/automation-practice-form/");

driver.manage().timeouts().implicitlyWait(30, TimeUnit.***SECONDS***);

WebElement CaptureText=driver.findElement(By.*xpath*("//h1[contains(text(),'Automation Practice Form')]"));

System.***out***.println("We are using getText method to print the webelement of text=> "+CaptureText.getText());

WebElement AttributeText=driver.findElement(By.*name*("firstname"));

AttributeText.sendKeys("selenium");

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

System.***out***.println(AttributeText.getAttribute("innerHTML"));

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

System.***out***.println(AttributeText.getAttribute("value")+" <= We are using getAttribute method to print the webelement of value attribute");

System.***out***.println(AttributeText.getAttribute("type")+" <= We are using getAttribute method to print the webelement of type attribute");

System.***out***.println(AttributeText.getAttribute("name")+" <= We are using getAttribute method to print the webelement of name attribute");

WebElement AttributeText1=driver.findElement(By.*tagName*("body"));

System.***out***.println(AttributeText1.getText());

}

}

What is difference between Implicit, Explicit and fluent wait with syntax?

Note: Selenium does not provide by default synchronization we have to use in the form of implicit, explicit and fluent wait.

Note: We should never use implicit wait and explicit wait together Not Recommended because Selenium WD will wait first IMPLICIT WAIT then EXPLICIT WAIT will be applied, so both time will be added or increase together for every element.

NOTE: Difference between implicit wait Vs explicit wait?

1. Implicit wait always applied for the entire element with in the specified webpage, but explicit wait applied only specific element.
2. Implicit wait is available in the form of implicit keyword or method, but explicit wait is available in the form of web driver wait with some expected condition.
3. In implicit wait do not need to pass the driver into the constructor but in explicit wait we have to pass the driver into the constructor.

Note: Both waits are in dynamic nature.

* driver.manage().timeouts().implicitlyWait(2, TimeUnit.***SECONDS***);

I

implicit wait is a dynamic wait, this wait will wait until unless the entire web element is fully loaded on the webpage within given time, if given time is expired or exceed then implicit wait will be failed and if entire web element is fully loaded before within given time then remaining time will be ignored.

* driver.manage().timeouts().pageLoadTimeout(2, TimeUnit.***SECONDS***);

Pageloadtimeout is a dynamic wait, this pageloadtimeout method is wait until page is not fully loaded within given time, if given time is expired or exceed then pageloadtimeout method will be failed and if page is fully loaded before within given time then remaining time will be ignored.

It is again an extension of Implicit wait, but it used in page loading. It sets the amount of time Selenium to wait for a page load to complete before throwing an error. If the timeout is negative, page loads can be indefinite.

WebDriver driver;

@Test

**public** **void** f() {

System.*setProperty*("webdriver.chrome.driver","D:\\Tools\\chromedriver\_win32\\chromedriver.exe");

driver=**new** ChromeDriver();

driver.get("http://www.half.ebay.com");

driver.manage().timeouts().implicitlyWait(2, TimeUnit.***SECONDS***);

driver.manage().timeouts().pageLoadTimeout(2, TimeUnit.***SECONDS***);

WebElement AttributeText=driver.findElement(By.*name*("\_nkw"));

AttributeText.sendKeys("selenium");

System.***out***.println(AttributeText.getAttribute("type"));

}

}

* WebDriverWait wait= **new** WebDriverWait(driver, 3);

WebElement text=wait.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("//p[text()='QTP']")));

text.isDisplayed();

Explicit wait is a dynamic wait, this wait will wait until unless the particular web element is fully loaded, or particular condition is satisfy on the webpage within given time, if given time is expired or exceed then explicit wait will be failed and if particular web element is fully loaded or particular condition is satisfy before within given time then remaining time will be ignored.

How to take screenshot?

By using getscreenshotAs method, we can take screenshot.

**public** **class** TakeScreenShot {

WebDriver driver;

@Test

**public** **void** f() {

System.*setProperty*("webdriver.chrome.driver","D:\\Tools\\chromedriver\_win32\\chromedriver.exe");

driver=**new** ChromeDriver();

driver.get("http://www.google.com");

File src=((TakesScreenshot)driver).getScreenshotAs(OutputType.***FILE***);

FileUtils.*copyFile*(src, **new** File("D:\\Users\\guptaav\\Practise\\com.practise\\src\\test\\java\\Selenium\\google.png"));

}

}

What is DOM(document object model) and how many types of DOM?

DOM(document object model) is API or interface which is provided by specific browser. When a web page loaded that time browser created DOM page wise at runtime. DOM changed to page wise means each page has own DOM.

There are 3 types of DOM:-

1. HTML DOM(Based on html document)
2. XML DOM(based on xml document)
3. CORE DOM(based on other document)

CRUD:

Create: Add

Retrieve: Get

Update: Modify

Delete: Delete

NOTE:- Selenium always interacts with HTML DOM

What is the property of HTML DOM?

What are the names of selenium Web driver Exception?

1. No Such Element Exception – this exception occurs when we provided wrong locator by id, name etc., or that web element is not available on the specific page and that web element is not fully loaded because of page is not fully loaded.
2. ElementNotVisibleException – this exception occurs the element is present in the DOM but not visible, Visibility means the height and width should be greater than zero. Hidden Elements are defined in HTML using of type=”hidden” in the end of the html tag.
3. stale Element reference Exception – this exception occurs the element is deleted or no longer exist on the DOM of the page because of when we refresh the page that element is removed from catch memory because whenever page is loaded or refresh catch memory also deleted then will get this exception.
4. Timeout Exception – this exception occurs when page is not fully loaded or continually loaded until given time.
5. Web driver Exception – this exception occurs could be any reason may be selenium does not support specific browser version or driver version.
6. No Alert present exception – this exception occurs when we written alert code but alert is not present on the screen.
7. NoSuchFrameException- this exception occurs when we written frame code but frame is not present on the screen.
8. NoSuchWindowException- this exception occurs when we written switch to window code but window is not present on the screen.
9. NoSuchSessionException, SessionNotFoundException – this exception occurs web driver is finding to element but suddenly browser is closed driver.quit
10. RemoteDriverServerException – this exception occurs when we are using grid and connection could not be established because of reason.
11. IllegalStateException – When we haven’t set the browser driver or browser driver path is not set.

What is Iframes in selenium and how to identify the Iframe inside the HTML page and how to handle and Why should not use index in frame and how to count total no of frames?

* In selenium Iframes is treated as a web element, other web element is also there inside the frames.
* Just click on right click on mouse and click on view page source then you will get available frames in the specific page.
* We can handle by using frame index, frame name in the form of String “frame name” and frame web element.
* By using List<WebElement> Totalframe=*driver*.findElements(By.*tagName*("frame"));

System.***out***.println("Totalframe :"+Totalframe.size());

**public** **class** Frame\_Handling {

**static** WebDriver *driver*;

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

System.*setProperty*("webdriver.chrome.driver","D:\\Tools\\chromedriver\_win32\\chromedriver.exe");

*driver*=**new** ChromeDriver();

*driver*.get("https://www.freecrm.com/");

*driver*.manage().timeouts().implicitlyWait(30, TimeUnit.***SECONDS***);

WebElement username=*driver*.findElement(By.*name*("username"));

username.sendKeys("naveenk");

WebElement password=*driver*.findElement(By.*name*("password"));

password.sendKeys("test@123");

Thread.*sleep*(3000);

WebElement login=*driver*.findElement(By.*xpath*("//input[@value='Login']"));

login.click();

List<WebElement> Totalframe=*driver*.findElements(By.*tagName*("frame"));

System.***out***.println("Totalframe on this webpage :"+Totalframe.size());

WebElement framename=*driver*.findElement(By.*name*("mainpanel"));

//driver.switchTo().frame(1);

//driver.switchTo().frame("mainpanel");

*driver*.switchTo().frame(framename);

*driver*.findElement(By.*xpath*("//a[text()='Contacts']")).click();

}

}

How to print Total\_No\_Of\_Links\_RadioButton\_CheckBox\_ Dropdowns are available in web application?

**public** **class** Total\_No\_Of\_Links\_RadioButton\_CheckBox\_Dropdowns {

**static** WebDriver *driver*;

**public** **static** **void** main(String[] args) {

System.*setProperty*("webdriver.chrome.driver","D:\\Tools\\chromedriver\_win32\\chromedriver.exe");

*driver*=**new** ChromeDriver();

*driver*.get("http://seleniumpractise.blogspot.in/");

List<WebElement> TotalLinks=*driver*.findElements(By.*tagName*("a"));

System.***out***.println("TotalLinks on this webpage :"+TotalLinks.size());

List<WebElement> TotalRadiobutton=*driver*.findElements(By.*xpath*("//input[@type='radio']"));

System.***out***.println("TotalRadiobutton on this webpage :"+TotalRadiobutton.size());

List<WebElement> TotalCheckbox=*driver*.findElements(By.*xpath*("//input[@type='checkbox']"));

System.***out***.println("TotalCheckbox on this webpage :"+TotalCheckbox.size());

List<WebElement> TotalDropdowns=*driver*.findElements(By.*tagName*("select"));

System.***out***.println("TotalDropdown on this webpage :"+TotalDropdowns.size());

}

}

**public** **class** Test1 {

**static** WebDriver *driver*;

**static** WebElement *ele*;

**public** **static** **void** main(String[] args) {

System.*setProperty*("webdriver.chrome.driver","D:\\Tools\\chromedriver\_win32\\chromedriver.exe");

*driver*=**new** ChromeDriver();

*driver*.get("http://seleniumpractise.blogspot.in/");

List<WebElement> TotalLinks=*driver*.findElements(By.*tagName*("a"));

System.***out***.println("TotalLinks on this webpage :"+TotalLinks.size());

**for**(WebElement element:TotalLinks)

{

String LinkName= element.getText();

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

//element.findElement(By.xpath("//a[contains(text(),'Click this link to start new Tab')]")).click();

}

*ele*=*driver*.findElement(By.*xpath*("//a[contains(text(),'Click this link to start new Tab')]"));

*ele*.click();

}

}

**public** **class** Test2 {

**static** WebDriver *driver*;

**public** **static** **void** main(String[] args) {

System.*setProperty*("webdriver.chrome.driver","D:\\Tools\\chromedriver\_win32\\chromedriver.exe");

*driver*=**new** ChromeDriver();

*driver*.get("file:///D:/check.html");

List<WebElement> TotalCheckbox=*driver*.findElements(By.*xpath*("//input[@type='checkbox']"));

System.***out***.println("TotalCheckbox on this webpage :"+TotalCheckbox.size());

**for**(**int** i=0;i<TotalCheckbox.size();i++)

{

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

TotalCheckbox.get(i).click();

String CheckboxName= TotalCheckbox.get(i).getAttribute("type");

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

}

}

}

How to generate log file in selenium and java using log4j?

Logs means what is going on at the time of program execution, so we can say all activities is captured while execution time and debugging the code if failure occurs.

* What things we need to do
* We have to put one defined code in file with properties extension and with name log4j.
* Based on requirement we can log4j.appender.file.Append=false here if you keep false then your previous logs will be over write(deleted) only new logs will be displayed if you keep true pervious logs also displayed.
* Wherever you want to use log4j the we have to create one reference variable of logger class then call the logger class and select one method which is available in logger class .get logger in that method we have to pass class name with .class
* After previous step where you want use in whole class we have to call, that reference variable of logger class then select method like: info, warn method you can use it and in that method we have to pass only string values and you will get that logs line or message in between of two lines of scripts while execution.
* If you are not declared logger code in class where you want to generate logs by default logs will be generated if you want customize logs then you can use info, warn method etc. using reference variable of logger class.

NOTE – if in log4j file log4j.appender.console=org.apache.log4j.ConsoleAppender this line will be commented log will not be generated Even though your logger code are declared in your class.

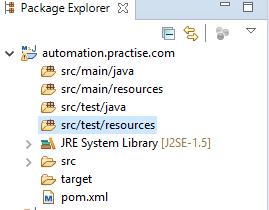
What thing we needed for install maven?

First of all You have to download and install latest JDK.

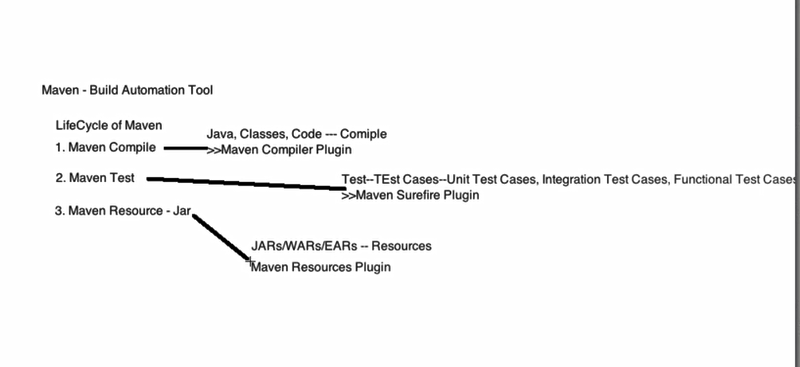
How to install maven in eclipse?

In eclipse by default maven available, you just go file menu click on file menu and go there in new and select project then you have to search maven in search box then you will get maven project and click on maven project and you have to click on create a simple project check ox and click on next then you have to give group id and artifact id in content box then you will get maven project structure whatever you given in artifact id by default that is your project name.

* Once you have created maven project successfully then you will get below maven structure



Maven life cycle?



1. In maven compile, we can run all the classes in one shot just like testng.xml through maven compiler plugin and we have to put in Pom.Xml.
2. In maven test, we can run all the test cases, which is available in the class with @test annotation if you not declared @test annotation might be that test cases will not run, through maven surefire plugin and we have to put in Pom.Xml.

NOTE –

1. In maven, you can run your test cases with TestNG u can do it without TestNG also.
2. If your test cases failed then maven build will be failed.
3. In maven resource jar, we can create jar/war/EARs through maven resource plugin and we have to put in Pom.Xml, the benefit is we can share the jar/war/Ears to other team members which located in other place or client, so they can run our classes and only .class files will be share not .java files.

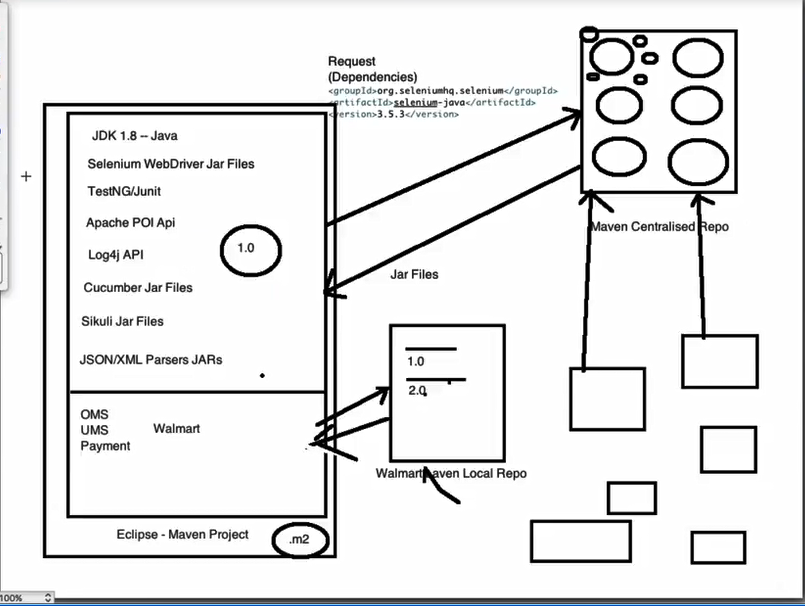
What is maven and advantages?

Maven is the build automation tool based on the concept of project object model (POM)

Its contains information of project and configuration information such as a dependencies and plugin etc.

Advantages -:

1. We can add dependency instead of jars.
2. No need to update the dependency again2 automatically it will update the dependencies from central repository instead of jars.
3. We can run the test cases from CMD through maven.
4. Maven provide .m2 folder where all plugins and dependencies stores in our local system for eclipse project.
5. We can create our local repository.



And major advantage is by maven pom.Xml we can create, own group id and artifact id with version name and upload to central repository, so another location team member can access our dependency, so advantage is another location team member do not need to ask to our team which dependency or jar file you are using just tell them you can access our dependency by group id, artifact id with version name.

In which location .m2 folder created in system?

Ans. C:\users\systemName\.m2

What will be there in .m2 folder in system?

Ans. In .m2 folder where all plugins and dependencies stores in our local system.

What is Juint, advantages, and disadvantages?

How many type of annotation in Junit and use?

Write a code for dropdown by List <Web element> or collection?

Write a code for textbox by List <Web element> or collection?

Taking screenshot for only failed test cases ?

1. First of all, we have create some test cases in test case class and those test cases should be failed explicitly and extends to take screen shot base class and declare @Listeners(CustomListener.**class**)
2. Now we must create one base class in that class we to declare takescreeshot method.
3. Now we have to create one class(CustomListener) in that class we have to extends ItestListener interface and extends take screen shot base class and implement all the method of Itestlistener interface and in all of them one method is there that is Ontestfailure in that method we have declare failure method

NOTE: Whenever any test cases has failed who will listen @Listeners(CustomListener.**class**) will listen and execute that onTestfailure which is implemented and in that method failure method will also excute to take the screenshot.

How to run your failed test cases multiple time?

What is the difference between POI and JXL for read and write by excel?

What is object repository and how many types of object repository in selenium and use?

What is the use of get Attribute and get Text?

How to close the browser in selenium?

* Driver. Close();//It closes the the browser window on which the focus is set.
* Driver. Quit();//It basically calls driver.dispose method which in turn closes all the browser windows and ends the WebDriver session gracefully.

How many types of exception in selenium?

How many types of find element in selenium and use?

How to handle dynamic calendar and web table?

How to use Robot class in Selenium?

What is web driver listener and how to implement listener in selenium web driver?

How to capture tooltip in selenium?

How to get current system date and time in selenium?

How to run fail test cases automatically in selenium?

How to hidden web element in selenium web driver?

How to download file in selenium?

What is java script and what is use of java script in selenium?

How to create [Database connection- Database Testing in Selenium using MS Access and Oracle](http://learn-automation.com/database-testing-in-selenium-with-jdbc-odbc-using-oracle-and-ms-access/)

How to use extent report?

How to perform headless browser testing using HTML UNIT DRIVER and why should use?

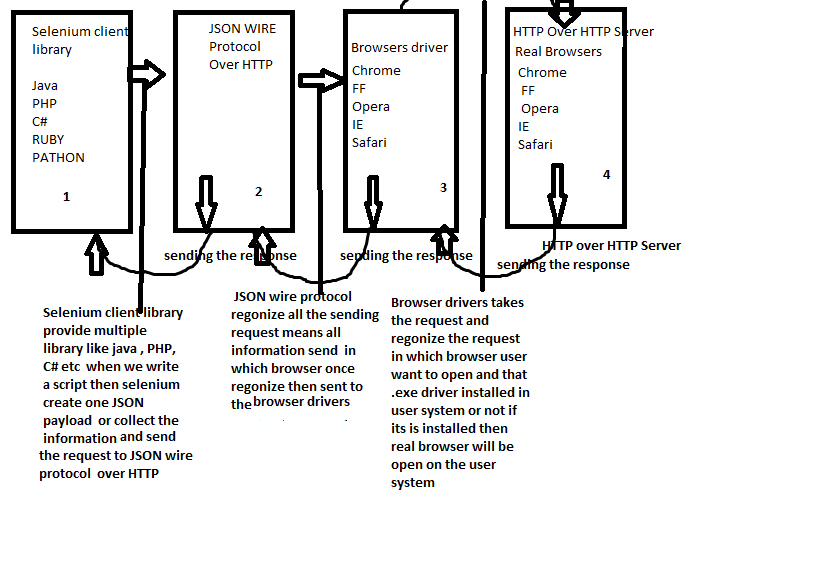
How to perform headless browser in HTML UNIT DRIVER for java script test?

What things we cannot automatein HTML UNIT DRIVER?

How to perform headless browser testing using PHANTOM JS and why should use?

How to integrate selenium with Jenkins and use of Jenkins?

What is selenium architecture and how it works/ How Selenium WebDriver works? **OR**What is architecture of Selenium WebDriver? OR Why do we require browser drivers?



* In selenium everything is available in the form API and in selenium arch. Total 4 components are there which would be external means user cannot see that and which would be internal means user can see that and all components are working parallelly.
* Selenium Client library: Selenium client library provides multiple library or programming languages in form of jar file and API, so whenever write the selenium script then selenium client library will create one JSON pay load file and put all the information in that and send to the JSON wire protocol over the HTTP.
* JSON wire protocol: JSON wire protocol accept all the request which is sent by selenium client library and recognize all the information send to which browser once they recognize send to the respective browser driver.
* Browser drivers: Browser driver accept all the information which is sent by JSON wire protocol and recognize in which browser user want open and that respective browser .exe file is installed or not on user system if it is installed then send to the respective browser driver
* Real browser: Real browser will be open.
* If something is going wrong on real browser which means any request or response is wrong then It will send the response in the form of API code 200, 500 etc. in the same process which earlier sent the request parallelly but won’t show the error code 200, 500 etc to the client or user then selenium convert the API code into selenium error code to the client or user.

12. What is selenium grid?

-selenium grid is just like a configuration if you want run the test cases on multiple OS and multiple browsers or different system with the help of the remote web driver and multiple desired capabilities we can execute our test cases.

Selenium grid provides mechanism where we can run our automation test cases in distributed environment against multiple O.S and as well multiple browsers, it works as a hub and node combination where multiple node can connect to a single hub.

#what things are needed for selenium grid to run the test cases?

-We need to create one-hub and node servers.

1. Hub is just like a server or that machine where you have written all the test scripts and hub will connect to node through JSON Wire protocol/HTTP request.

Hub is server or virtual machine that drives test execution and distributes to the nodes with required configuration. Code execution is done on the hub side. Hub send the commands to the nodes to perform action on the browser through JSON Wire protocol/HTTP request.

2. Node is just like a server or that machine or VM where you want to run your test cases on multiple OS as well as multiple browsers.

-Only one hub to be create.

-Multiple nodes can be create.

#How to create a Hub or start the hub server?

1. We have to download selenium server standlone jar file in hub machine.

2. Go to cmd just go into that directory or folder where you have kept the selenium server standlone jar file in hub machine.

STEP ->

1.Go to selenium server standlone jar file directory or folder

2. Type java -jar and path of selenium server standlone file with extension -role hub,s (java -jar selenium-server

standalone-3.8.1.jar -role hub ) and press enter button.

3.Then you should see in cmd two lines <Nodes should register to http://192.168.0.3:4445/grid/register and selenium Grid hub is up and running now you can assume, you have successfully created hub server in hub machine.

4.Here you can check this hub is really up or not just open browser copy this URL //192.168.0.3:4445 and paste it in browser

then if you are getting Selenium Grid Hub v.3.8.1 page, now you can assume, you have successfully created hub server in hub machine.

NOTE -> If you are getting exception during STEP 2 (java.net.BindException: Address already in use: bind) then you have to use another port or use this command (java -jar selenium-server standalone-3.8.1.jar -role hub -port 4445) and press enter button.

#How to create a Node or start the node server?

1.We have to download selenium server standlone jar file in node machine.

2.We have to download any browser or where you want to excute your test cases

3.We have to download related browser driver

4.Go to cmd just go into that directory or folder where you have kept the selenium server standlone jar file in node machine.

STEP ->

1.Go to selenium server standlone jar file directory or folder

2.Type java -Dwebdriver.chrome.driver="C:\Users\guptaav\Downloads\chromedriver\_win32\chromedriver.exe" -jar selenium-server-standalone-3.8.1.jar -role node -hub http://10.141.233.96:4444/grid/register/

3.Then you should see in cmd four lines Launching a Selenium Grid node, Selenium Grid node is up and ready to register to the hub, Registering the node to the hub: http://192.168.0.3:4445/grid/register and The node is registered to the hub and ready to use, now you can assume, you have successfully created node server in node machine.

4.Then you should see in hub server "Registered a node http://192.168.0.3:5555"(Just copy this url and paste in the browser)

#We have to write a code in eclipse platform for launching the browser with the help of DesiredCapabilities and new RemoteWebDriver and code in below.

DesiredCapabilities cap=new DesiredCapabilities();

cap.setPlatform(Platform.WIN10);

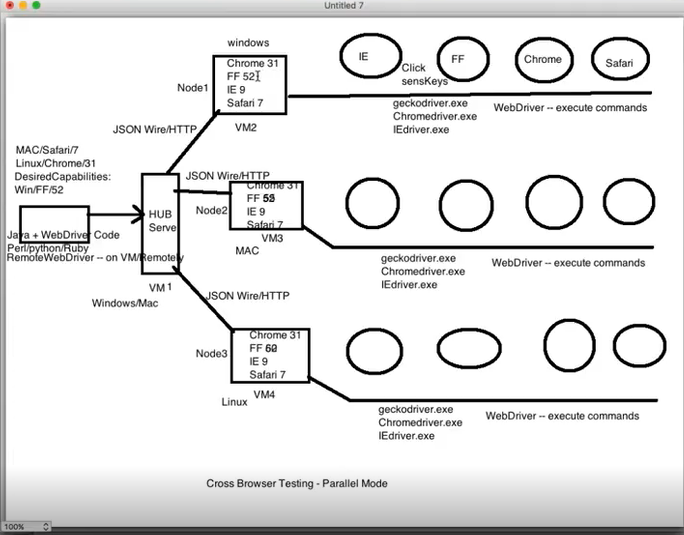
cap.setBrowserName("chrome");

WebDriver driver=new RemoteWebDriver(new URL(<http://192.168.0.3:4445/wd/hub>,cap), cap);//here you have to give that URL that is recived on hub Machine.

driver.get("http://www.google.com");

System.out.println(driver.getTitle());

14. #how to upload jar or dependency file or maven local repo in central repo?



29. What is properties file and how to create properties file in selenium ?

* In Selenium .properties files are mainly used to store GUI locators / object repository, test data, and also global fields like URL, browserName details etc and in .properties file data will be store in the form key and value format
* First of all we have to create one .properties file

NOTE – 1. If something is commented in properties file then will be print null on console.

2. If in properties file is mentioned key name different and in class key name is different then will be print null on console.

Ex:

URL=http://toolsqa.com/automation-practice-form/

AttributeText=firstname

selenium=selenium

value=value

type=type

name=name

* After that we have to create one java class

Ex:

**public** **class** ReadPropFile {

**static** WebDriver *driver*;

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

// **TODO** Auto-generated method stub

Properties pro=**new** Properties();

FileInputStream fis=**new** FileInputStream("D:\\Users\\guptaav\\Practise\\com.practise\\src\\test\\java\\Selenium\\Properties.properties");

pro.load(fis);

System.*setProperty*("webdriver.chrome.driver","D:\\Tools\\chromedriver\_win32\\chromedriver.exe");

*driver*=**new** ChromeDriver();

System.***out***.println(pro.getProperty("URL")+" => coming From properties file");

*driver*.get(pro.getProperty("URL"));

WebElement CaptureText=*driver*.findElement(By.*xpath*("//h1[contains(text(),'Automation Practice Form')]"));

System.***out***.println("We are using getText method to print the webelement of text=> "+CaptureText.getText());

WebElement AttributeText=*driver*.findElement(By.*name*(pro.getProperty("AttributeText")));

AttributeText.sendKeys(pro.getProperty("selenium"));

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

System.***out***.println(AttributeText.getAttribute("innerHTML"));

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

System.***out***.println(AttributeText.getAttribute(pro.getProperty("value"))+" <= We are using getAttribute method to print the webelement of value attribute");

System.***out***.println(AttributeText.getAttribute(pro.getProperty("type"))+" <= We are using getAttribute method to print the webelement of type attribute");

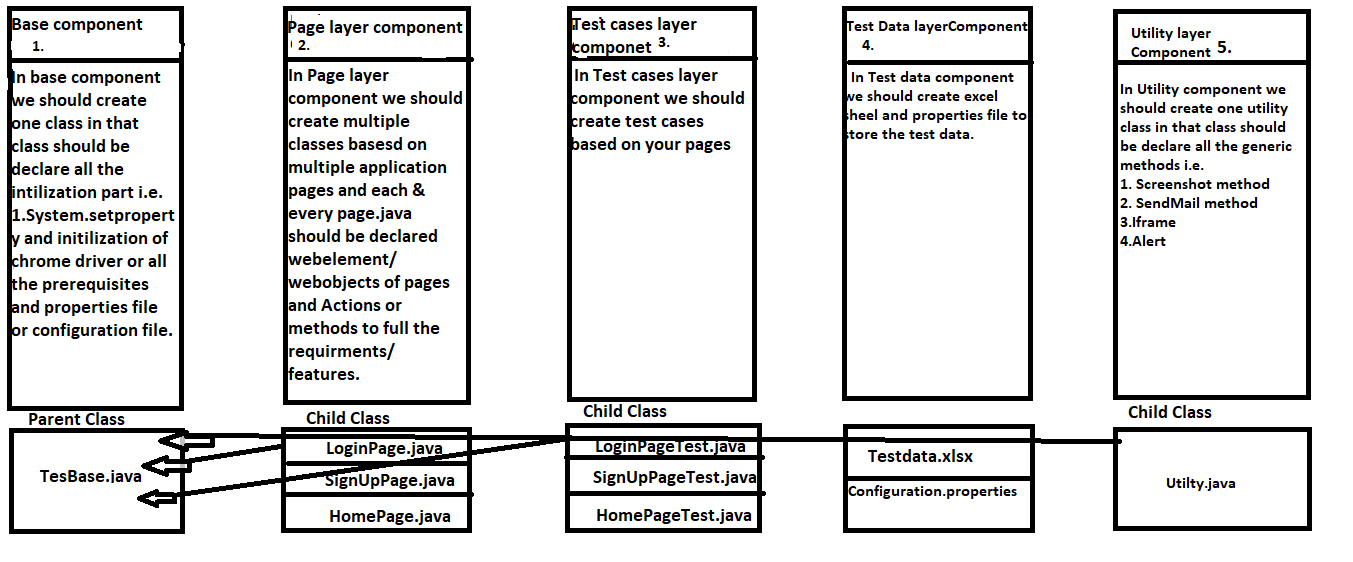
System.***out***.println(AttributeText.getAttribute(pro.getProperty("name"))+" <= We are using getAttribute method to print the webelement of name attribute");

}

}

What is framework?

Framework is set of guidelines or rules where write as much as automation code systematically in efficient way, less maintenance and code reusability These guidelines could include coding standards, test-data handling methods, object repositories, processes for storing the test results, and utilities.



* Page object model is pattern or approach where we have to create the separate java class for each and every web page, where we have to declare initialization of web driver, methods/actions, web elements/web objects, test cases, test data and utilities based on web pages, so In page object model multiple components are there and all component are available in the form of packages and classes.
* Test Base component: In test base component we have to create one package under the package create one class in that class we have to declare all the initialization part i.e. Web driver initialization, system.setproperty, properties file or configuration file and all the prerequisites etc.
* Page layer component: In page layer component we have to create multiple package and java classes as well based on web page and in each every java class, we have to declare methods/ action and web elements/ web objects in the form of java methods.
* Test case layer component: In Test case layer component we have to create multiple package and java classes as well and in each and every java class, we have to write the test cases based on page layer component.
* Test data layer component: In Test data layer component we have to create multiple excel file and properties file to store the test data based on web pages.
* Utility layer component: In utility component, we have to declare generic functions or methods where we can use at the time of writing the test cases i.e. Take screenshot, email, frames alerts methods etc.
* NOTE: Here Test base component is parent class of all remaining classes and each and every child class have to extend the property of parent class by inheritance concept.

\*Component content is available in the each component diagram.

Advantages:

* 1. Code reusability: Here we can save the time no need to write the same property in each class, we can achieve code reusability with the help of inheritance concept.
  2. Code maintainability: Here all things i.e. Initialization of prerequisites, page wise specific code, test cases code, generic methods and test data are declare individually, so we can modify easily based on requirement at any time.
  3. Object repository: Here all the web elements are declare page wise.