**SELENIUM WBDRIVER WITH CORE JAVA BASICS**

**Install java jdk -> download ecipse ->import selenium jarfiles**

**What makes selenium unique from other tools?**

1. It is open source automation tool and supports different browsers, os and languages and for web based apps.

**Web Architecture?**

The editor and the web site which is to be tested are directly interacted with its browsers native language to talk.

**Different versions of selenium?**

Selenium RC 3.0.1

Selenium WebDriver 3.0.1

**Browser support**

IE, Google Chrome, Firefox, Safari

**Languages**

Java, c#, python, ruby, Javascript (node), perl , php

**Os support –** windows, OS x Linux, Solaris

**Components of Seleium**

**Latest version – selenium(3.2.0)**

**selenium IDE** - having a record and playback options which is present in every automation tool like qtp

**selenium rc** - injects javascript into browsers when the page is loaded

**selenium webdriver** - automation tool to test web apps in diff browsers using diff languages

**selenium grid**- Selenium-Grid is used to speed up the execution of a test pass by using multiple machines to run tests in parallel.

**selenium core**-same like selenium webdriver but Selenium Core tests run directly in a browser

**Why not older version of selenium RC?**

When a user sends a command through rc -> the proxy server in the middle interprets it and then it triggers the java script of the test site according to the request.

1) In today’s world this web browsers became robust and they are not allowing the triggering (due to hacking or some other issues)

2) RC is not compatible to some api’s

3)even we need to start the proxy server manually running the command line.

So, in order to get rid of java script events triggering they avoided older version.

**What are the tools in general used to fing the objects for particular application?**

Firebug, Firepath - Firefox

Chrome inspect - chrome

F12 – ie

**Selenium locators & which one to select**

Xpath

Id, Css

Name, Class name, Tag Name

Linktext Partial Linktext,

* It depends on their uniqueness that is ill choose which one is robust and ill select that.

**Css locator syntax :** tagName[attribute=’value’] (or)

Id - #value Class - .value (if you see spaces in value, replace with “.”)

**Choosing class object reference with reference to the interface?**

Invoking the driver class

System.setproperty(“key”,”value”);

**Key** – [webdriver.gecko.driver(firefox)](http://www.gecko.driver(firefox))

[webdriver.chrome.driver(chrome)](http://www.chrome.driver(chrome))

[webdriver.ie.driver(ie)](http://www.ie.driver(ie))

**value** – exe file path

1.select the browser driver class ex: FirefoxDriver()

2. Create an object for that

d=new FireFoxDriver();

3. then make class object reference to the WebDriver interface

WebDriver d = new FireFoxDriver();

**WebDriver Interface?**

A WebDriver(interface) is a set of methods but it won’t implement them. A class should be their to implement all the methods defined in that interface.

**What are the classes u are using to implement the methods present in WebDriver?**

**Depends on the browser choice**

AndroidDriver, AndroidWebDriver

ChromeDriver, FirefoxDriver

InternetExplorerDriver, HtmlUnitDriver, EventFiringWebDriver

IPhoneDriver, IPhoneSimulatorDriver, RemoteWebDriver, SafariDriver

**How to handle without using Firepath that is using with our own customised Xpath?**

Syntax - .//tagName[@attribute=’value’]

**Traversing tag names**

**->**in this instead of using the required field we can actually take the whole reg box which belongs to it and traverse slowly from top to that point. **Example:**

.//tagName[@attribute=’value’]/tagName/tagName

If it contains 2 child - > give tag name

* div[1]

**Imp points before coding**

**->verify in firebug before u run.**

**->use css customized path**

**->id - #**

**->class - .**

**->find the unique attribute**

**Web elements validation:**

**Isdisplayed-** for checking whether

the radio or check is visible or hidden

**isSelected –** to see whether radio or checkbox is selected as per our request.

* If something is not present in the webpage then inorder to chech that use “.size” and check with the count of that element by giving if condition
* If you are navigating from home to contact or any page then use above approach to check whether ur in that particular page or not

**isEnabled :** to check whether the option is enable or disable

**Assertions:**They will check whether the script is running as expected

* **assertTrue** – script runs only if the condition in assert is true
* **assertFalse** – script runs only if the condition in assert is false

**Dynamic dropdown:** which gives suggestions base on your entry

**Static dropdown:** which is constant

* Default select method will be there and create an object for it then access by using that object – value, index and visibleText.

**Checkboxes:** can be taken directly

**Radiobutton:** write for loops and if else conditions wherever possible to check dynamically.

**Alerts: 2 types**

1. written in java language (use alert method)

-> driver.switchTo().alert().

sendkeys/getText/accept()/dismiss()

2) written with html attributes (just go with normal xpath approach)

**Thread.sleep(5000L)->To wait for 5 sec**

**Handling Multiple Windows in selenium?**

**By default webdriver shows the page which was mentioned then in order to switch to other window we need to specify by saying**

**Driver.switchTo().window(“pass id”);**

**getWindowHandles()** will give u the opened windows and their id’s (parent, child and subchild windows)

-> through all ids into **Set<String> datastructure.**

**Set<String> ids = driver.getWindowHandles();**

**->**then traverse into it with the help of **iterator** method

Iterator<String> it = ids.iterator();

->place parent and child id into a string

String parentid = it.next();

String childid = it.next();

-> place that id into the Driver.switchTo().window(childid);

**Frames: (very imp)**

These are html docs which are embedded on other html docs

* Developer uses iframe or frameset

driver.switchTo().frame(id)

**How do you handle dropdown in selenium webdriver?**

Using select class and picking the methods i.e.., by index, value, visibletext and so on

**How do you handle java alerts?**

**Alert is the method we need to switch**

-> driver.switchTo().alert().

sendkeys/getText/accept()-OK/dismiss()-CANCEL

**how do you handle checkboxes?**

**IsSelected method – checked or unchecked**

**IsDisabled method –** true or false(Boolean)

**IsEnabled method –** true or false(Boolean)

**If I want to select radio button. Unfortunately the 3 radio buttons have same names then how do u handle it?**

I use find element and get all those attributes then based on index I traverse and I will click on to that particular button.

**How do u handle drag and drop, caplock, rightclick, hover to particular ui ….,**

By using actions class

**For AJAX controllers:** (mouse hover ,capslock, rightclick, drag and drop i.e., all external operations)

Do it by using **action** class and u can get all the relative methods in it

->for all action methods add build().perform() at the end

object.moveToElement(xpath).build().perform

**Screenshots** are helpful in crossbrowser testing and it tells the flow of the application whether it is working correctly or not

**Taking Screenshots:**

It can be done by using TakesScreenshot interface. Here It cannot create object and also implement methods so we have to **typecast** driver class for this

TakesScreenshot ts = (TakesScreenshot)driver;

Now, we have take screenshot type

ts.getScreenshotAs(OutputType,FILE);

Now we got screenshot in the form of a file and we have to store that in destination file with the help of **FileUtils**

FileUtils.copyFile(source,destination);

**Tracking failed scripts:**

**ITestResult** is an interface which tracks the execution of test cases. We can get status of the test case and test case name.

ITestResult.FAILURE/SUCCESS/CREATED/SKIP/STARTED

**Assert and Verify commands:**

**TestNG has Assert and no Verify**

**Assert-**it will fail the test and abort the current test case execution. It is best used to make sure your test case is gng in the right direction.

**Verify-**it will fail the test and continue to run the test case. It is best used to compare your test results with actual outcomes

**Creating Java Project and set up database:**

**Step-1: Download java project and import in Eclipse IDE**

* Click file and select open Projects from file systems -> click on directory -> select the project.
* Now download jdbc driver from <https://dev.mysql.com/downloads/connector/j/> and import in the lib folder of the project i.e.., right click on project -> click on properties -> select Java Build Path -> go to libraries -> add external jar files and import jar file.
* Now for giving connection for the jdbc driver give username and password as root and root. Database is connected successfully.

**Step-2: Import .sql files into mySql database**

* At first, create database in mysql file for that select 4th icon which is create new schema in the connected server.
* For importing .sql files we have to go to the server and select Data import -> select Import from Self-contained file and browse .sql file from computer -> set target destination (choose database which is given in the java file) for Default target schema -> start import -> tables will be imported into the database.

