Selenium:

1.open source automation tool

2.supports multiple languages like ,java,python

3.supports/automates only web applications/browser application or url based appln.

4.supports multiple browsers.

5.thoughtworks company take cares of selenium

Disadv:

4.does not support desktop application like,skype,goto meeting

Major components:

1.record and play: selenium IDE

2.selenium RC :not using now.

3.selenium webdriver: latest version is 3.0.4

Main competitor of selenium is QTP

QTP:

Licensed tool,have to pay and supported by HP

Disadv:

Supports only VB script,runs oly on windows.

Note: both JAR files and library are same

We have two libraries:

1.JRE System library:

Jars supporting JAVA

2.Refernced libs:

Jars for selenium and any other tool or anything like apache POI API

Launching different browsers in selenium:

1.firefox browser:

To launch we need to use firefox driver

Webdriver driver =new firefoxDriver();

🡪firefoxdriver is a class and we are creating an object for the class and referring it to an interface Webdriver.

🡪Webdriver – is an interface

🡪relationship between webdriver interface and firefix driver class is predefined by selenium.

Setting property to driver :

System.setproperty(”key”,”value”);

🡪set property is available in system method

Validation :

🡪we can use equals

e.g.if(title.equals(“google”))

{}

Else

{}

Locators in selenium:

How to identify html tag of a web element?

Whatever written next to bracket,is its html tag

<input id=””>🡪input is the html tag

All the links are represented by <a > tag

What is xpath?

Address of a particular element

Note:

Preference:

1.by.id

2.by.xpath : Absolute xpath shd not be used(hierarchy xpath)

Relative xpath only should be used.

3.by.name

For links: by.linktext

Absolute xpath:it starts with single slash and it starts from root node.

If any changes happens in path of web element,test fails.

Relative xpath:

Starts with middle of html DOM,starts with double slash,which means it can search any web element from any part o web page

Xpath axes:

Used to find out complex or dynamic elements which does not have a proper xpath id,classname or name etc.

Axes methods:

Child,parent,ancestor,sibling,preceeding,self

Syntax:

1.following method:

Selects all elements in the current node.

//\*[@type=’text’]//following::input

2.Ancestor:selects all ancestor elements[parent,grandparents]

//\*[]@type=’’]//ancestor::div

--lists all matching parent nodes,to sleect particular index,use //\*[]@type=’’]//ancestor::div[1]

3.child: only single slash

Child elements of current node

//\*[@type=’text’]/child::li

4.preceding:all nodes that comes before current node

//\*[]@type=’’]//preceding::div

5.following-sibling:siblings of current node

//\*[@type='submit']//following-sibling::input

6.parent:parent of current node

//\*[@id='rt-feature']//parent::div

### 7. Descendant

Selects the descendants of the current node as shown in the below screen.

In the below expression, it identifies all the element descendants to current element ( 'Main body surround' frame element) which means down under the node (child node , grandchild node, etc.).

//\*[@id='rt-feature']//descendant::a

Dropdown:

1.use select class,since we need to select particular element

2.create object for select class

e.g. WebElement e=driver.findElement(By.*id*("inputCountry"));

Select s=**new** Select (e);

s.selectByVisibleText(country);

Thread.*sleep*(5000);

Findelements:

Consider the example:

We need to count links in a page and write the text of all links

How to do:

With the help of findelements

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

System.***out***.println(ll.size());

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

{

String s=ll.get(i).getText();

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

}

findElements --🡪For identifying more than one webelement,.

List<WebElement>🡪for storing the list of webelements

ll.size());-🡪to get the size of link i.e.count

for loop-🡪to print the text of elements

ll.get(i).getText();

--🡪ll.get(i)🡪gets all the webelements in the list

ll.get(i).getText();🡪prints all the text of webelements in the list

Alerts/Pop-ups:

how to handle alerts/confirmation pop-ups/javascript pop-ups?

driver.get("https://mail.rediff.com/cgi-bin/login.cgi");

driver.findElement(By.*name*("proceed")).click();

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

alt.accept();

Thread.*sleep*(5000);

driver.quit();

//First,we need to switch the control from main window to pop-up,so

driver.switchTo().alert();🡪used to switch the control from main window to pop-up

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

Alert popup is available in Alert class,so import and create an obj reference.

alt.accept();🡪to click ok button in alert pop-up

alt.dismiss();🡪to click no button in pop-up

alt.gettext()🡪to get the alert message in the pop-up

File upload pop-up/window pop-up:

Note:selenium does not support any window pop-up/local machine where file is saved.

it supports only javascript pop-up,which is a part of browser

driver.get("https://html.com/input-type-file/");

driver.findElement(By.*id*("fileupload")).sendKeys("C:\\Users\\Raj\\Desktop\\daily\_task.txt");

note:for file upload,we need to use our local disc,and it cannot be accessed by selenium.

So,

1.first,find upload button,but donot click it,bcz,control will go to local disc,instead,use sendkeys and give the location of the file,and file will be uploaded.

Frame handling in selenium:

Frame is nothing but webpage within a webpage

2 important properties of frames:

1.id and name

i.e. id:u van pass frame index

name:u can pass name=””

driver.get("http://www.londonfreelance.org/courses/frames/index.html");

Thread.*sleep*(5000);

driver.switchTo().frame("navbar");

driver.findElement(By.*linkText*("No frames")).click();

driver.switchTo().frame("navbar");

🡪control switches from page to frame ,here,we r passing frame(String)i.e.name of frame

🡪after this,u can click element inside the frame