**Selenium components**

IDE – plugin (TestNG comes under plugin)

RC server - complex, slow. (you -> selenium RC server -> Browser) Web driver is called as RC

Web driver - (you -> Browser) it is a API and Interface

Grid – it is used automate remote machines

API – Application programming interface

Java API – it contains 🡪Packages 🡪 classes 🡪 methods (all are default)

WebDriver API – it contains 🡪 Packages 🡪 classes 🡪 methods (Packages them in jar files)

Firebug 🡪 it is add on for firefox browser used for displaying html script for the webpage and highlighting the element

FirePath 🡪 generating Xpath automatically

**Creating WebDriver Project in Eclipse:**

Creating java project

Download jar files

Associate jar files to java projects

**How to launch the web browser using web driver?**

System.*setProperty*("webdriver.edge.driver", "C:\\Users\\keerthiraja.sp\\Downloads\\edgedriver\_win64\\msedgedriver.exe");

EdgeDriver driver =new EdgeDriver(); //driver constructor used to invoke the browser. It is constructor to create object class for the web driver

WebDriver driver =new EdgeDriver();

// webdriver is interface EdgeDriver is instant of class

// it is invoke the empty browser

driver.get("https://www.tutorialspoint.com/index.htm");

//it is used to give the instruction to invoke web page to same bowser

**Methods:**

findElement() 🡪 This method is used to finding the element on web page. This function returns the first element found with the provided locator strategy used.

findElements() 🡪 This function returns all the elements matched for the given locator strategy.

**Locator:**

**Selenium IDE:** A component of selenium (Record and playback tool) plugin for firefox browser

HTML/DHTML/XML 🡪 client side scripting language

**types of locators:**

it will provide address of the element. Id, Name, Link text, CSS Selector and xpath.

**1)Id locator:**

Syntax:

Driver.findElement(By.id("Email")).sendKeys(“keerthiraja@gmail.com”);

**2)Name locator:**

Syntax:

Driver.findElement(By.name("username")).sendKeys(“keerthiraja”);

**3)class Name locator:**

Syntax:

Driver.findElement(By.className("Email")).sendKeys(“keerthiraja@gmail.com”);

**4)Link Text locator:**

Syntax:

Driver.findElement(By.linkText("Register")).click();

**5)CSS Selector:**

Tag and ID, Tag and Class, Tag and Attribute, (Tag, attribute, class) and Inner text

**i)Tag and Id combination:**

<input type="checkbox" id="checkbox1" value="Cricket">

Here input is a tag name and id is a checkbox1. When we use Tag with id combination we must mention hashtag (#) symbol for the id. By.cssSelector("Input#id")

Syntax:

Driver.findElement(By.cssSelector("input#checkbox1")).click();

**ii)Tag and Class combination:**

<input type="tel" class="form-control" ng-model="Phone" required pattern="^\d{10}$">

Here input is a tag name and classname is a form-control. When we use Tag with class combination we must mention dot (.) symbol for the class name. By.cssSelector("Input.classname")

Syntax:

Driver.findElement(By.cssSelector("input.form-control")).sendKeys("9578631352");

**iii)Tag and attribute combination:**

<input type="text" placeholder="Last Name" name="LastName" class="form-control" ng-model="LastName" required>

Here input is a tag name and attribute is anything else of the property like Classname, id, name. When we use Tag with attribute combination we must mention squire bracket ([]) symbol for the attribute. By.cssSelector("Input[name=lastName]")

Syntax:

Driver.findElement(By.cssSelector("input[name=lastName]")).sendKeys("Keerthiraja");

**iv)Tag, Class and Attribute:**

<input type="text" placeholder="Last Name" name="LastName" class="form-control" ng-model="LastName" required>

Here we use tag, class and attribute to find the web element. When we use all combination we must mention tag name and class is a dot (.) and attribute is a squire bracket ([]) symbol. By.cssSelector("Input.classname[name=lastName]")

Syntax:

Driver.findElement(By.cssSelector("input.form-control[name=lastName]")).sendKeys("Keerthiraja");

**v)Inner text:**

when the object doesn’t have id, name, or class, attributes, then we use of their inner texts. Inner texts are actual string patters that the html label shows on the page.

<font size="2" face="Arial, Helveticas, sans-serif">Password</font>

Here inner text is password. So, we identify inner text must use colon (:) symbol for the inner text CSS method.

Syntax:

Driver.findElement(By.cssSelector("font:contains("Password")"));

**6)xpath locator:**

xpath is basically address of web element. It is used to easily identify the web elements.

Absolute Xpath – Complete or full address of the element in the web page.

Relative Xpath – partial address of the element is the web page.

**i)Absolute Xpath:**

absolute xpath always begins with the single forward slash (/), which means you can select the element from the root node.

driver.findElement(By.xpath("/html/body/section/div/div/main/div/div/div/div/div/div/table/tbody/tr/td[1]/table/tbody/tr/td[2]/div/div/ol/li"));

**ii)Relative Xpath:**

Relative Xpath always starts with double slash (//). Always preferred relative Xpath because absolute xpath maybe change by the developer during maintenance time. So it is not safe for using absolute xpath so, relative xpath always preferable.

Write the xpath manually by using //tagname[@attribute=’value’]

driver.findElement(By.*xpath*("//\*[@id='credit2']/a"));

**Using Xpath handling complex & Dynamic elements in selenium:**

**1)Basic Xpath:**

Xpath expression select nodes or list of nodes on the basics of attributes like Id, Name, Classname etc.

**2)Using OR & AND:**

In OR expression, two conditions are used, whether 1st condition OR 2nd condition should be true.

Syntax:

Xpath=//\*[@type=’submit’ OR @name=’btnReset’]

In AND expression, two conditions are used, both conditions should be true to find element. It fails to find element if any one condition is false.

Syntax:

Xpath=//\*[@type=’submit’ AND @name=’btnLogin’]

**3)Using single attribute:**

Syntax:

//tagname[@attribute-name=’value’]

Example:

//a[@href=’https://www.google.com’]

//input[@id=’name’]

//input[@name=’username’]

//img[@alt=’sometext’]

**4)Using multiple attribute:**

Syntax:

//tagname[@attribute1=’value1’] [@attribute2=’value2’]

Example:

//a[@id=’id’][@name=’namevalue’]

//img[@src=’’][@href=’’]

**5)Using contains method:**

we can handle elements whose property dynamically changing. It is matches all the passible dynamically changing values.

Syntax:

//tagname[contains(@attribute-name,’value’)]

🡪matches email email123 email321 femail ahgemail dynamically changed values.

**Example:**

//input[contains(@id,’email’)]

//input[contains(@name,’email’)]

//a[contains(@href,’’)]

//img[contains(@src,’’)]

//div[contains(@id,’’)]

**6)Using starts-with method:**

Name=’submit005’ but does not match dynamically changing value it will matches only stating with in the mentioned values.

Syntax:

//tagname[starts-with(@attribute-name,’value’)]

Example:

//input[starts-with(@name,’submit’)]

🡪 matches submit001, submit872, submit-827

**WebDriver commands:**

**Get commands:**

get() 🡪 it is used to open the url on the browser

getTitle() 🡪 it is used to get title of page

getCurrentUrl() 🡪 returns url

getText() 🡪 return text value of an element

**Browser commands:**

close() 🡪 close a current window

quit() 🡪 closes multiple windows

**Navigation commands:**

Navigate().forward() 🡪 used to move forward of the page

Navigate().back()🡪 used to move backward of the page

Navigate().to() 🡪 it used to navigate to the page provided in the function.

Navigate().refresh() 🡪 it is used to refresh the current page

**Conditional commands:**

isDisplayed() 🡪 It is used to display the particular web element present or not in the given web element

isEnabled() 🡪 It is used to check the condition about it is enabled or not in the given web element

isSelected() 🡪 It is used to check the condition about radio button or check box is selected or not in the given web element

**Switch commands:**

switchTo().frame() 🡪 this method are used to switch one frame to another frame

switchTo().alert() 🡪 accept(), dismiss()

switchTo().defaultContent() 🡪 Default command is used to come out of the previous web element

switchTo().window() 🡪 this method are used to switch one window to another window

WindowHandles() 🡪 this method are used to handle multiple windows on same time

WindowHandle() 🡪 this method are used to handle single windows on same time

**wait commands:**

Implicit wait 🡪 implicit wait is a global wait command

Explicit wait 🡪 explicit wait command is wait particular web element

**important web elements:**

textbox/input Box, button, checkbox, Dropdown menu, Radio button, Link, images, Calendars (date pickers), Web table, File upload, frame, download file.

**1)Text box/ Input box:**

Text box is used to enter some values in the text field. here we are follow sendkeys() method to pass the value on the text field.

driver.findElement(By.*name*("identifier")).sendKeys("keerthiraja1999@gmail.com");

**2)Button:**

Button is used to perform submit or click action. Here we are using submit or click() method. Mostly click method are used.

driver.findElement(By.*className*("gb\_E")).click();

**3)check box:**

Check box is used to select multiple option. Here we are using click method to enable the option.

**4)Dropdown menu:**

Drop down menu is a set of list and used to select one option on the listed element. Here we used select () method to select the list.

There are three methods is used to find the position of list.

selectByValue("Engineering");

selectByVisibleText("HTML");

selectByIndex (5);

**5)Radio button:**

Radio button is used to select single option like true or false. Here we used click () method to enable the option.

isSelected ();

isDisplayed ();

isEnabled ();

**How to find Broken links?**

A broken link or dead link is a link on a web page that on longer works because the website is encountering one or more of the reasons below.

An improper URL entered for the link by the website owner.

The destination website removed the linked web page (causing what is known as a 404 error).

The destination website permanently moved or no longer exists

Brocken links can be problematic for website visitors, making them usable to access the desired resource or information.

**Data driven tests:**

Reading the data from excel sheet.

**Download apache poi package:**

<https://www.apache.org/dyn/closer.lua/poi/release/bin/poi-bin-3.16-20170419.zip>

download all jars and add to java project build path.

**Flow of excel:**

Excel 🡪 workbook 🡪 sheets 🡪 rows 🡪 cells

**FileInputStreem Class:**

FileInputStream 🡪 it is used to open excel file from the path

XSSFWorkbook 🡪 it is used to create work book on the excel sheet

XSSFSheet 🡪 it is used to return the sheet from the workbook

XSSFRow 🡪 it is used to return the rows from the sheets

XSSFCell 🡪 it is used to return the cells from the rows

**Method for stream class:**

toString () method is a generic method which is used to read the all type of data.

getStringCellValue ();

getNumaricCellValue ();

getBooleanCellValue ();

sendkeys () method is not allowed while passing numeric values. But, String.value() method used to pass the value side the parameter of sendkeys

sendkeys(12345); // not allowed

sendkeys(String.valueOf(12345)); //allowed

convert String to integer 🡪 Integer.parseInt();

convert String to double 🡪 Double.parseDouble();

**Robot class:**

In certain selenium automation tests, there is a need to control keyboard or mouse to interact with OS windows like download pop-up, alert, print pop-ups, etc.

Selenium web driver cannot handle these OS- pop-ups/applications.

**Benefits of robot class:**

Robot class can simulate keyboard and mouse events.

Robot class can help in upload/download of files when using selenium web driver.

Robot class can easily be integrated with current automation frame work (keyboard, data-driven or hybrid).

**Common use cases using robot class:**

To press down arrow key of keyboard we use

Robot.keypress(KeyEvent.VK\_DOWN)

To press TAB key of keyboard we use

Robot.keypress(KeyEvent.VK\_TAB)

To press Enter key of keyboard we use

Robot.keypress(KeyEvent.VK\_ENTER)

**Selenium Commands for cookies:**

Return the list of all cookies.

Driver.manage().getCookies();

Return specific cookie according to name.

Driver.manage().getCookieNamed(arg0);

Create and add the cookies.

Driver.manage().addCookie(arg0);

Delete specific cookies.

Driver.manage().deleteCookie(arg0);

Delete specific cookie according name.

Driver.manage().deleteCookieNamed(arg0);

Delete all cookies.

Driver.manage().deleteAllCookies(arg0);

**Page object model:**

Organizing the page elements.

Driver.findElement(By.name(“username”)).sendkeys(“Keerthiraja”);

findElement(By.name(“username”)) 🡪 Identification

sendkeys(); 🡪 operational method

**Actions we can perform using JavaScript executor:**

flashing an element

drawing boarder around the element

capture title of the page

click in some element

generate alert info

refreshing page

scrolling page

**Database testing:**

SQL🡪 Structured query language

DDL 🡪 create, alter, drop

DML 🡪 select, insert, update, delete

TCL 🡪 commit, rollback

DCL 🡪 grant, revoke

**JDBC:**

Download database drivers (MySQL connector) and associate to your project

Steps to connect database and execute queries through java program:

create connection

create statement

execute statement

close connection

**Selenium Grid:**

Grid is a component of selenium used to executing the test cases remotely.

Hub (Master) 🡪 One hub

Node (Slave) 🡪 Many

**Pre-requisites:**

we have drivers & bowsers on local & remote systems.

make local system as HUB server, then register all the VM’s (nodes) with Hub.

Convert local system as a hub server: (run the code in local machine)

Open command prompt and type the following command.

🡺java –jar selenium-server-standalone-3.5.3.jar –role hub

The above command is used to make it as a hub server of a machine. After type the command it return as node registration link.

🡺Nodes should register to <http://10.157.216.79:4444/grid/register/>

Check hub is running or not:

🡺https://localhost:4444/grid/console

Convert local system as nodes: (run the code in remote machine)

🡺Java –Dwebdriver.edge.driver=”C:\Users\keerthiraja.sp\Downloads\edgedriver\_win64\msedgedriver.exe” –jar selenium-server-standalone-3.5.3.jar –role node –hub <http://10.157.216.79:4444/grid/register/>

download selenium standalone jars and we should have same jar on all the nodes.

write teste case then executes on remote systems.

Windows(hub) 🡪 node(windowsOs and Edge)

**TestNG:**

**What is Assertion?**

Assertion helps us to verify the conditions of the test and decide whether test has failed or passed. A test is considered successful only if it is completed without throwing exception.

Types of assertion:

1. Hard assertion
2. Soft assertion

**i) Hard assertion:**

Assert.assertTrue();

Assert.assertFalse();

Assert.assertEquals();

testNg - <http://dl.bintray.com/testng-team/testng-eclipse>