**Cross Browser Testing in Selenium:**

1:Cross browser testing using selenium Grid.

Selenium grid can be used to execute testcases in different browser, in different machines, in different browser versions, in different OS simultaneously. It dramatically accelerates the testing process across browsers and across platforms by giving us quick and accurate feedback.

Selenium Grid allows us to execute multiple instances of WebDriver or Selenium Remote Control tests in parallel which uses the same code base, hence the code need NOT be present on the system they execute.

Selenium Grid has a Hub and a Node.

* **Hub** - The hub can also be understood as a server which acts as the central point where the tests would be triggered. A Selenium Grid has only one Hub and it is launched on a single machine once.
* **Node** - Nodes are the Selenium instances that are attached to the Hub which execute the tests. There can be one or more nodes in a grid which can be of any OS and can contain any of the Selenium supported browsers. Node can be in same system different port or in different system

Steps:

1: Download selenium standalone server (this is a jar file).

2: Execute the downloaded jar from cmd to start the server as hub

java -jar selenium-server-standalone-2.52.0.jar -port 4444 -role hub

By default hub will start in port 4444 if we don’t pass while executing the jar.

3: Check if the hub started by going to <http://localhost:4444/grid/console>

4: Execute the jar again in a cmd to start the server as node

java -jar D:\JAR\selenium-server-standalone-2.42.2.jar -role node -hub http://10.30.217.157:4444/grid/register -browser browserName=firefox -port 5555

Where 10.30.217.157=hub’s ip address

By default node will start in port 5555 if we don’t pass while executing the jar.

Set path for chrome/id driver

-Dwebdriver.ie.driver=”path”

-Dwebdriver.chrome.driver=”path”

5: Execute the jar again in a cmd to start the server as node chrome driver

java -jar D:\JAR\selenium-server-standalone-2.42.2.jar -role node -hub http://10.30.217.157:4444/grid/register -browser browserName=chrome -port 5556 -Dwebdriver.chrome.driver=”chrome driver path”

Where 10.30.217.157=hub’s ip address

6: Selenium script:

public class TestNGClass

{

public WebDriver driver;

public DesiredCapabilities cap;

public String node;

@Parameters({"browser"})

@BeforeTest

public void launchBrowser String browser) throws MalformedURLException

{

if (browser.equalsIgnoreCase("firefox"))

{

System.out.println(" Executing on FireFox");

cap = DesiredCapabilities.firefox();

cap.setBrowserName("firefox");

node=" http://firefox\_node\_ipaddress:5555/wd/hub";

} else if (browser.equalsIgnoreCase("chrome"))

{

System.out.println(" Executing on Chrome");

cap = DesiredCapabilities.chrome();

cap.setBrowserName("chrome");

node=" http:// chrome\_node\_ipaddress:5556/wd/hub";

}

driver = new RemoteWebDriver(new URL(node), cap);

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

}

@Test

public void goToURL()throws Exception {

driver.navigate().to(“http://google.co.in”);

Thread.sleep(5000);

}

@AfterTest

public void quitBrowser(){

driver.quit();

}

7: The contents of the XML file are shown below. We create 2 tests and put them in a suite and mention parallel="tests" so that all the tests would be executed in parallel.

Testing.xml:

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name="Suite" parallel="tests">

<test name="FirefoxTest">

<parameter name="browser" value="firefox" />

<classes>

<class name="TestNG.TestNGClass" />

</classes>

</test>

<test name="ChromeTest">

<parameter name="browser" value="chrome" />

<classes>

<class name="TestNG.TestNGClass" />

</classes>

</test>

</suite>

2: Cross browser testing in same machine(can be done without using grid)

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.Parameters;

**import** org.testng.annotations.Test;

**public** **class** TestBase {

/\*

\* **Crossbrowser testing in selenium** can be done to execute cases simultaneously on more than one browser in same machine, (if we want to execute in different machine we have to use grid) in testng.xml file we need to set the parallel attribute to execute the cases parallel in different browser

\* Set seperate test blocks in testng.xml for seperate browser value passed as parameter and set parallel=tests so that all test blocks are executed simultaneously

\*/

**public** WebDriver driver;

@Parameters({"browserName"})

@BeforeMethod

**public** **void** launchBrowser(String browser){

**if**(browser.equalsIgnoreCase("firefox")){

driver=**new** FirefoxDriver();

}**else** **if**(browser.equalsIgnoreCase("chrome")){

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\Rupjit\\Desktop\\Desktop\\selenium-java-2.44.0\\chromedriver\_win32\\chromedriver.exe");

driver=**new** ChromeDriver();

}**else**{

System.*out*.println("Incorrect browser name passed so launching default browser-Firefox");

driver=**new** FirefoxDriver();

}

}

@AfterMethod

**public** **void** quitBrowser(){

driver.quit();

}

@Test

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

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

Thread.*sleep*(5000);

}

}

Testng.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name=*"Suite"* parallel=*"tests"*>

<test name=*"Test-Firefox"*>

<parameter name=*"browserName"* value=*"firefox"*></parameter>

<classes>

<class name=*"pack1.TestBase"*/>

</classes>

</test>

<test name=*"Test-Chrome"*>

<parameter name=*"browserName"* value=*"chrome"*></parameter>

<classes>

<class name=*"pack1.TestBase"*/>

</classes>

</test>

</suite>