Selenium:[Creating test suite using Junit and eclipse in selenium](http://selenium-suresh.blogspot.com/2012/08/creating-test-suite-using-junit-and.html)

**How to create test suite using Junit and eclipse in selenium**  
  
There are some scenarios where we need to run multiple test cases. Either we can run those test cases independently or together. But there are some real time cases where we need to run our test cases in a particular order. In this case we would prefer Test Suite to combine the test cases together and decide their orders and run those.  
**Below are the steps**  
1. Create a Test Suite class where we create the Test Suites which will call all the test cases in a particular order.  
import junit.framework.Test;  
import junit.framework.TestCase;  
import junit.framework.TestSuite;   
import junit.textui.TestRunner;  
public class TestSuite1 extends TestCase {  
public static Test suite()   
{   
TestSuite suite = new TestSuite();   
suite.addTestSuite( TestCase1.class);   
suite.addTestSuite( TestCase2.class);  
suite.addTestSuite( TestCase3.class);   
return suite;   
}   
public static void main(String arg[])  
 { TestRunner.run(suite()); }  
}  
Step 2. Create your first test case   
import org.openqa.selenium.server.RemoteControlConfiguration;  
import org.openqa.selenium.server.SeleniumServer;  
import com.thoughtworks.selenium.\*;  
public class TestCase1 extends SeleneseTestCase{  
Selenium selenium;  
public static final String MAX\_WAIT\_TIME\_IN\_MS="60000";  
private SeleniumServer seleniumServer;  
public void setUp() throws Exception {  
RemoteControlConfiguration rc = new RemoteControlConfiguration();  
rc.setSingleWindow(true);  
seleniumServer = new SeleniumServer(rc);  
selenium = new DefaultSelenium("localhost", 4444, "\*iexplore", "http://www.google.com/");  
seleniumServer.start();  
selenium.start();  
}  
public void testgoogling() {  
selenium.open("/");  
selenium.type("q", "Suresh");  
selenium.click("btnG");  
selenium.waitForPageToLoad(MAX\_WAIT\_TIME\_IN\_MS);  
assertTrue(selenium.isTextPresent("Niraj"));  
}  
public void tearDown() throws InterruptedException{  
selenium.stop();   
seleniumServer.stop();  
}  
}  
Step 3. Create your second test case   
import org.openqa.selenium.server.RemoteControlConfiguration;  
import org.openqa.selenium.server.SeleniumServer;  
import com.thoughtworks.selenium.\*;  
public class TestCase2 extends SeleneseTestCase{  
Selenium selenium;  
public static final String MAX\_WAIT\_TIME\_IN\_MS="60000";  
private SeleniumServer seleniumServer;  
public void setUp() throws Exception {  
RemoteControlConfiguration rc = new RemoteControlConfiguration();  
rc.setSingleWindow(true);  
seleniumServer = new SeleniumServer(rc);  
selenium = new DefaultSelenium("localhost", 4444, "\*iexplore", "http://www.google.com/");  
seleniumServer.start();  
selenium.start();  
}  
public void testgoogling() {  
selenium.open("/");  
selenium.type("q", "Suresh Babu");  
selenium.click("btnG");  
selenium.waitForPageToLoad(MAX\_WAIT\_TIME\_IN\_MS);  
assertTrue(selenium.isTextPresent("Suresh Babu"));  
}  
public void tearDown() throws InterruptedException{  
selenium.stop();   
seleniumServer.stop();  
}  
}  
Step 4. Create your third test case   
import org.openqa.selenium.server.RemoteControlConfiguration;  
import org.openqa.selenium.server.SeleniumServer;  
import com.thoughtworks.selenium.\*;  
public class TestCase3 extends SeleneseTestCase{  
Selenium selenium;  
public static final String MAX\_WAIT\_TIME\_IN\_MS="60000";  
private SeleniumServer seleniumServer;  
public void setUp() throws Exception {  
RemoteControlConfiguration rc = new RemoteControlConfiguration();  
rc.setSingleWindow(true);  
seleniumServer = new SeleniumServer(rc);  
selenium = new DefaultSelenium("localhost", 4444, "\*iexplore", "http://www.google.com/");  
seleniumServer.start();  
selenium.start();  
}  
public void testgoogling() {  
selenium.open("/");  
selenium.type("q", "http:/selenium-suresh.blogspot.com");  
selenium.click("btnG");  
selenium.waitForPageToLoad(MAX\_WAIT\_TIME\_IN\_MS);  
assertTrue(selenium.isTextPresent("[http://selenium-suresh.blogspot.com](http://selenium-suresh.blogspot.com/)"));  
}  
public void tearDown() throws InterruptedException{  
selenium.stop();   
seleniumServer.stop();   
}  
}  
Step 5. Run your Test Suite   
Go to your Test Suite class and right click on that and run as Junit Test.  
This will run the TestCase1 then TestCase2 then TestCase3  
If you want to execute your test cases in some specific order then you call them in that order like.  
suite.addTestSuite( TestCase3.class);   
suite.addTestSuite( TestCase2.class);  
suite.addTestSuite( TestCase1.class);   
Above will run the TestCase3 first then TestCase2 then TestCase1.

[using functions in xpath in selenium](http://selenium-suresh.blogspot.com/2012/08/using-functions-in-xpath-in-selenium.html)

**How to use functions in xpath in selenium**  
  
Automation using selenium is a great experience. It provides many way to identif an object or element on the web page.  
But sometime we face the problems of idenfying the objects on a page which have same attributes. When we get more than  
one element which are same in attribute and name like multiple checkboxes with same name and same id. More than one button having  
same name and ids. There are no way to distingues those element. In this case we have problem to instruct selenium to identify a perticular  
object on a web page.  
I am giving you a simple example . In the below html source there are 6 checkboxes are there having same type and same name.  
It is really tough to select third or fifth.  
input type='checkbox' name='chk' first

input type='checkbox' name='chk' second  
input type='checkbox' name='chk' third  
input type='checkbox' name='chk' forth  
input type='checkbox' name='chk' fifth  
input type='checkbox' name='chk' sixth  
Thare are some function we can use in Xpath to identify the abject in above cases. An XPath expression can return one of four basic XPath data types:

\* String \* Number \* Boolean \* Node-set  
XPath Type : Functions  
Node set : last(), position(), count(), id(), local-name(), namespace-uri(), name()  
String : string(), concat(), starts-with(), contains(), substring-before(), substring-after(), substring(), string-length(), normalize-space(), translate()  
Boolean : boolean(), not(), true(), false(), lang()  
Number : number(), sum(), floor(), ceiling(), round()  
I will show you how we can use some of these above functions in xpath to identify the objects.  
Node Set : last()  
In the above html file there are six checkboxes and all are having same attributes (same type and name)  
How we can select the last checkbox based on the position. We can use last() function to indentify the last object among all similar objects.  
Below code will check or uncheck the last checkbox.  
selenium.click("xpath=(//input[@type='checkbox'])[last()]");  
How we can select the second last checkbox and third last checkbox. We can use last()- function to indentify the last object among all similar objects.  
Below code will check or uncheck the second last checkbox and thrid last checkbox respectively.  
selenium.click("xpath=(//input[@type='submit'])[last()-1]");  
selenium.click("xpath=(//input[@type='submit'])[last()-2]");  
Node Set : position()   
If you want to select any object based on their position using xpath then you can use position() function in xpath.  
You want to select second checkbox and forth checkbox then use below command  
selenium.click("xpath=(//input[@type='checkbox'])[position()=2]");  
selenium.click("xpath=(//input[@type='checkbox'])[position()=4]");  
above code will select second and forth checkbox respectively.  
String : starts-with()   
Many web sites create dynamic element on their web pages where Ids of the elements gets generated dynamically.  
Each time id gets generated differently. So to handle this situation we use some JavaScript functions.  
XPath: //button[starts-with(@id, 'continue-')]   
Sometimes an element gets identfied by a value that could be surrounded by other text, then contains function can be used.  
To demonstrate, the element can be located based on the ‘suggest’ class without having  
to couple it with the ‘top’ and ‘business’ classes using the following  
XPath: //input[contains(@class, 'suggest')].

[Identify dynamic element in selenium](http://selenium-suresh.blogspot.com/2012/08/identify-dynamic-element-in-selenium.html)

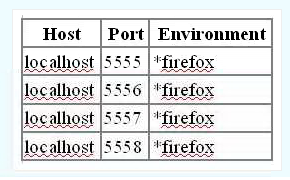
**How to identify dynamic element in selenium**  
  
Many web sites create dynamic element on their web pages where Ids of the elements gets generated dynamically. Each time id gets generated differently. So to handle this situation we use some JavaScript functions.  
**starts-with**   
if your dynamic element's ids have the format  where button id="continue-12345"  where 12345 is a dynamic number you could use the following  
**XPath:** //button[starts-with(@id, 'continue-')]   
**contains**   
Sometimes an element gets identfied by a value that could be surrounded by other text, then contains function can be used.  
To demonstrate, the element  can be located based on the ‘suggest’ class without having to couple it with the ‘top’ and ‘business’ classes using the following  
**XPath:** //input[contains(@class, 'suggest')].

[Selenium Installation](http://selenium-suresh.blogspot.com/2012/06/selenium-installation.html)

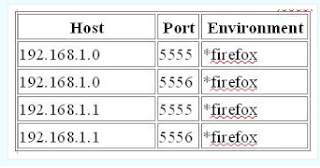
**Selenium IDE Installation**  
1. Download Selenium IDE plug in from seleniumhq.org site - [Click here to download SeleniumIDE](http://seleniumhq.org/download/)  
2. Open Firefox browser and do drag & drop   
3. Click on Install Now button   
4. Restart Firefox   
5. In Firefox Navigate to tools option here we can find Selenium IDE  
**Firebug Installation**  
1. Download Firebug plug in from Google  
2. Open Firefox browser and do drag & drop  
3. Click on Install Now button  
4. Restart Firefox   
5. In Firefox Navigate to tools option here we can find Firebug Tool  
**Configuring Selenium-RC with Eclipse**  
**Follow below steps to configure Selenium-RC with Eclipse:**  
Download Selenium RC java client driver from the Selenium-HQ ([Click Here to download Selenium RC](http://seleniumhq.org/download/)).  Extract selenium-java-.jar file   
Open your desired Java IDE (Eclipse)  
Create a java project.   
Configure Build Path with downloaded jar files to java project as references.   
**To configure build path** go to -->File -->Select build path -->Select Configure build path -->Select Libraries tab-->Click on Add External jars button -->Now add Selenium-Server.jar file which you saved in your system -->Click on OK .  
Run Selenium server from the console to execute your tests from the Java IDE (Eclipse)  
To run the selenium server from console - Open command prompt -->Navigate the path where you saved Selenium-server.jar file and type the command as >java -jar selenium-server.jar  
Eg: your jave file is saved in D drive in Selenium RC folder, then we need to use below command  
    D :>Selenium RC> java -jar selenium-server.jar  
**Selenium server will be started**

[Selenium Grid - Setup in System](http://selenium-suresh.blogspot.com/2012/09/selenium-grid-setup-in-system.html)

**Selenium Grid Setup**  
  
1.Check that Java is installed in your system by using the command “java -version”. If Java is installed it will give the output about Java Version that is installed in your system. Else download the Java from Java website (<http://www.java.com/en/download/index.jsp>).  
  
2.Check ANT is installed in your system by using the command “ant -version”.If Ant is installed it will give the output about Ant Version that is installed in your system. Else download the Apache Ant from Apache website (<http://ant.apache.org/>).  
3.Download Selenium Grid from Selenium Grid website(<http://selenium-grid.seleniumhq.org/download.html>) and unpack it.  
  
4.Go to the Unpacked directory of the Selenium grid using command prompt and type the command “ant launch-hub”. This will launch the Selenium Grid Hub on a default port “4444” to check whether the selenium grid is running. Type the URL <http://localhost:4444/console>  on your browser and check that you are able to see the selenium hub console on the browser.  
  
5.Now go to the same Selenium Grid directory using command prompt and type the command “ant launch-remote-control”. This will start the Selenium Grid Remote control on default port “5555” and environment “\*firefox” and register it to the Hub which you had earlier started.  
  
6. Now after starting the selenium remote control we will test the example provided with Selenium Grid for sequence testing in single machine. For this go to the Selnium grid folder from command prompt and type the command “ant run-in-demo-sequence”. This will execute the test example provided with selenium grid to be executed in sequence.  
  
Now you may had noticed in the setting up steps that I am running the test in sequence. You may think in your mind that, what's special in that? The same things can be achieved when we use a Selenium RC.Then here comes the important part of Selenium Grid.   
  
**Setting up Parallel execution on a single machine**  
  
1. Start the hub and one remote control in the same way as in the earlier setting up steps.  
2. Now go to the Selenium Grid folder using command prompt and type the command  
  
“ant -Dport =5556 launch-remote-control”  
  
This will start another remote control on port “5556” and register it to the local hub. If you want to register more remote control to the Hub. You can do that by starting multiple remote control but with different ports.  
  
3. To know the no. of registered remote control onto a hub go to the URL <http://localhost:4444/console>. It will give you a view as follows:

[](http://3.bp.blogspot.com/-1XBMOYO2Cto/UGVMv0Q24oI/AAAAAAAAAD8/xaU3x3I9OzU/s1600/HostNames.png)

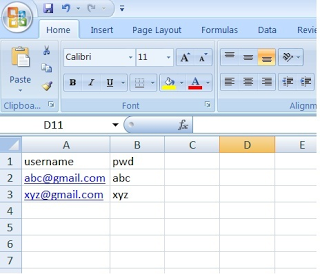
4.Now to run the test in parallel. Go to the selenium grid directory using command prompt and type the command “ant run-demo-in-parallel”. This will run the example tests given with selenium grid in parallel.  
  
**Setting up parallel execution for different machines and different environments**  
  
In this case start the Hub normally as you do in any of the system.   
  
For starting the remote control use the following command.  
  
ant -Dport= -Dhost= -DhubURL= -Denvironment= launch-remote-control  
Here  
“-Dport” should be used to set the port of the remote control.  
“-Dhost” should be used to set the host(where the remote control has been started) IP. This is the IP which the hub will use to talk to the remote control.  
“-DhubURL” should be used to set the Hub Url where the current remote control will register itself.  
“-Denvironment” should be used to set an environment name that remote control represents to the hub.  
       Now start 4 different remote controls from 2 different machines and register them to the hub.  
  After this check the Hub console. It will look similar to the table as shown below depending upon the remote control that are registered to the Hub.

[](http://1.bp.blogspot.com/-GzE3Q0yWFmc/UGVNUZvazfI/AAAAAAAAAEE/jjSMI5EGTYo/s1600/Host.png)

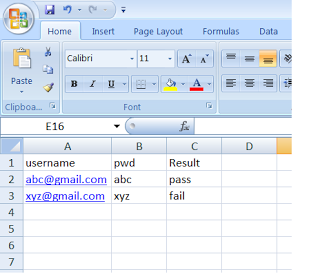
Then to verify your set-up go to the Selenium Grid folder from the command prompt and type the command “ant run-demo-in-parallel”  
  
This will run an example test accompanied with the Selenium Grid to run in parallel.   
  
**Advantages of Selenium Grid**  
  
1. Remote Execution of the selenium tests can be done using Selenium Grid without actually copying your test code to to remote system.  
2. Selenium Grid can be used to save time by executing independent testcases in parallel.  
3. Load Testing can be done if we register a lot of remote control to a single hub and execute testcase in parallel to simulate multiple transactions at a single timeperiod.

[Write results to excel sheet](http://selenium-suresh.blogspot.com/2012/07/write-results-to-excel-sheet.html)

**Write results to excel sheet**  
  
After successfully executing scripts, every one want to write results to excel sheet..here is the way to write results to excel sheet....  
Below is the sample script to write results to excel sheet...  
package test;  
import jxl.Sheet;  
import jxl.Workbook;  
import jxl.write.Label;  
import jxl.write.WritableSheet;  
import jxl.write.WritableWorkbook;  
import com.thoughtworks.selenium.\*;  
import org.openqa.selenium.server.\*;  
import org.testng.annotations.\*;  
  
public class Importexport1 {  
public Selenium selenium;  
public SeleniumServer seleniumserver;  
@BeforeClass  
public void setUp() throws Exception {  
RemoteControlConfiguration rc = new RemoteControlConfiguration();  
seleniumserver = new SeleniumServer(rc);  
selenium = new DefaultSelenium("localhost", 4444, "\*firefox", "http://");  
seleniumserver.start();  
selenium.start();  
}  
@Test  
public void testImportexport1() throws Exception {  
**// Read data from excel sheet**  
FileInputStream fi = new FileInputStream(  
"F:\\Framework\\testdata\\Login1\_Credentials.xls");  
Workbook w = Workbook.getWorkbook(fi);  
Sheet s = w.getSheet(0);  
String a[][] = new String[s.getRows()][s.getColumns()];  
**// Write the input data into another excel file**  
FileOutputStream fo = new FileOutputStream(  
"F:\\Framework\\Results\\LoginResult1.xls");  
WritableWorkbook wwb = Workbook.createWorkbook(fo);  
WritableSheet ws = wwb.createSheet("loginresult1", 0);  
selenium.open("http://www.gmail.com");  
selenium.windowMaximize();  
System.out.println("s.getRows() = " + s.getRows());  
for (int i = 0; i < s.getRows(); i++) {  
System.out.println("s.getColumns = " + s.getColumns());  
for (int j = 0; j < s.getColumns(); j++) {  
a[i][j] = s.getCell(j, i).getContents();  
Label l = new Label(j, i, a[i][j]);  
Label l1 = new Label(2, 0, "Result");  
ws.addCell(l);  
ws.addCell(l1);  
}}  
for (int i = 1; i < s.getRows(); i++) {  
selenium.type("Email", s.getCell(0, i).getContents());  
selenium.type("Passwd", s.getCell(1, i).getContents());  
selenium.click("signIn");  
selenium.waitForPageToLoad("30000");  
boolean aa = selenium.isTextPresent("The username or password you entered is incorrect. [?]");  
System.out.println("the value of aa is::" + aa);  
if (aa)  
{  
Label l3 = new Label(2, i, "fail");  
ws.addCell(l3);  
System.out.println("Login Failure");  
Thread.sleep(10000);  
} else {  
Label l2 = new Label(2, i, "pass");  
ws.addCell(l2);  
selenium.click("link=Sign out");  
Thread.sleep(10000);  
} }  
wwb.write();  
wwb.close();  
}  
@AfterClass  
public void tearDown() throws Exception {  
selenium.stop();  
seleniumserver.stop();  
}}  
Your input data should be like this....

[](http://4.bp.blogspot.com/-3Ff4BSZN7D0/UBeSMX9qgxI/AAAAAAAAADQ/uFDLwqm6pa0/s1600/Login1_Credentials.png)

Your out put excel should be like this

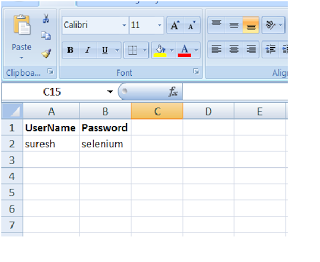
[](http://4.bp.blogspot.com/-PmKxtKZ13so/UBeRkslpAOI/AAAAAAAAADI/GvYCfw8MZrk/s1600/LoginResult1.png)

[Data Driven Testing Using Selenium RC with TestNG](http://selenium-suresh.blogspot.com/2012/07/data-driven-testing-using-selenium-rc_24.html)

How to do Data Driven Testing using TestNG   
There are many ways to do Data driven tests.I used Excel sheet for reading data and FileInputStream method.  
Below script explains you how to Read data from excel sheet and use the data to search google.  
I am using Eclipse, selenium RC, TestNG and Excel sheet for this script.  
  
--->Create a Java project in eclipse  
--->Create a package name as test  
--->Create a new class DatadriventestNG.  
--->Paste the below code in Eclipse.  
--->Chnage the path of excel file according to your requirement.  
Please note that selenium will support only .xls format plese do not forget to change the excel file to .xls if you are using MS-office2007.  
Below is the code  
  
package test;  
import jxl.Sheet;  
import jxl.Workbook;  
import com.thoughtworks.selenium.\*;  
import org.openqa.selenium.server.\*;  
import org.testng.annotations.\*;  
public class DefaultTNG {  
public Selenium selenium;  
public SeleniumServer seleniumserver;  
  @BeforeClass  
public void setUp() throws Exception {  
RemoteControlConfiguration rc = new RemoteControlConfiguration();  
seleniumserver = new SeleniumServer(rc);  
selenium = new DefaultSelenium("localhost", 4444, "\*iexplore", "http://");  
seleniumserver.start();  
selenium.start();  
}  
@Test  
public void testDefaultTNG()throws Exception {  
  
  FileInputStream fi=new FileInputStream("F:\\Framework\\testdata\\search.xls");  
Workbook w=Workbook.getWorkbook(fi);  
Sheet s=w.getSheet(0);  
selenium.open("<http://www.google.com/>");  
selenium.windowMaximize();  
for (int i = 1; i < s.getRows(); i++)  
 {   
//Read data from excel sheet  
 selenium.type("name=q",s.getCell(0,i).getContents());  
 selenium.click("btnG");  
 Thread.sleep(1000); }   
}  
@AfterClass  
 public void tearDown() throws InterruptedException{  
selenium.stop();  
seleniumserver.stop();  
    }}

[Login Scenario - Providing Login Detail from Excel Sheet](http://selenium-suresh.blogspot.com/2012/07/login-scenario-providing-login-detail.html)

**Excel Sheet format should be in .XLS format**

[](http://4.bp.blogspot.com/-yM2sYrVfXR0/UBDmC3NqLFI/AAAAAAAAAC0/SWELbdLlHeg/s1600/LoginScenario.png)

import java.io.FileInputStream;  
  
import jxl.Sheet;  
import jxl.Workbook;  
import com.thoughtworks.selenium.\*;  
import org.openqa.selenium.server.\*;  
import org.testng.annotations.\*;  
  
public class Loginexcel {  
public Selenium selenium;  
public SeleniumServer seleniumserver;  
  
@BeforeClass  
public void setUp() throws Exception {  
RemoteControlConfiguration rc = new RemoteControlConfiguration();  
seleniumserver = new SeleniumServer(rc);  
selenium = new DefaultSelenium("localhost", 4444, "\*firefox", "http://");  
seleniumserver.start();  
selenium.start();  
}  
  
@Test  
public void testDefaultTNG()throws Exception {  
FileInputStream fi=new FileInputStream("E:\\Selenium\\LoginExcel.xls");  
Workbook w=Workbook.getWorkbook(fi);  
Sheet s=w.getSheet(0);  
selenium.open("http://127.0.0.1/orangehrm-2.6/login.php");  
selenium.windowMaximize();  
for (int i = 1; i < s.getRows(); i++)  
{   
//Read data from excel sheet  
String s1 = s.getCell(0,i).getContents();  
String s2 = s.getCell(1,i).getContents();  
selenium.type("txtUserName",s1);  
selenium.type("txtPassword",s2);  
selenium.click("Submit");  
Thread.sleep(2000);   
selenium.click("Link=Logout");  
} }  
@AfterClass  
public void tearDown() throws InterruptedException{  
selenium.stop();  
seleniumserver.stop();  
}}

[Data Driven Testing using Selenium RC with JUnit](http://selenium-suresh.blogspot.com/2012/07/data-driven-testing-using-selenium-rc.html)

How to do Data Driven Testing using Selenium RC with JUnit  
There are many ways to do Data driven tests.I used Excel sheet for reading data and FileInputStream method.  
Below script explains you how to Read data from excel sheet and use the data to search google.  
I am using Eclipse, selenium RC, Junit and Excel sheet for this script.  
  
--->Create a Java project in eclipse.  
--->Create a new class DatadrivenJUnit.  
--->Paste the below code in Eclipse.  
--->Chnage the path of excel file according to your requirement.  
Please note that selenium will support only .xls format plese do not forget to change the excel file to .xls if you are using MS-office2007.  
Below is the code  
import java.io.FileInputStream;  
import jxl.Sheet;  
import jxl.Workbook;  
import org.junit.AfterClass;  
import org.junit.BeforeClass;  
import org.junit.Test;  
import org.openqa.selenium.server.RemoteControlConfiguration;  
import org.openqa.selenium.server.SeleniumServer;  
import com.thoughtworks.selenium.\*;  
  
public class DatadrivenJUnit extends SeleneseTestCase  
{  
public Selenium selenium;  
public SeleniumServer seleniumserver;  
@BeforeClass  
public void setUp() throws Exception   
{  
RemoteControlConfiguration rc = new RemoteControlConfiguration();  
rc.setSingleWindow(true);  
seleniumserver = new SeleniumServer(rc);  
selenium = new DefaultSelenium("localhost", 4444, "\*iexplore", "http://");  
seleniumserver.start();  
selenium.start();  
}  
@Test  
public void testDatadrivenJUnit() throws Exception  
{  
FileInputStream fi=new FileInputStream("F:\\Framework\\testdata\\search.xls");  
Workbook w=Workbook.getWorkbook(fi);  
Sheet s=w.getSheet(0);  
selenium.open("<http://www.google.com/>");  
selenium.windowMaximize();  
for (int i = 1; i < s.getRows(); i++)  
 { //Read data from excel sheet  
 selenium.type("name=q",s.getCell(0,i).getContents());  
 selenium.click("btnG");  
 Thread.sleep(1000); } }  
@AfterClass  
 public void tearDown() throws InterruptedException{  
selenium.stop();  
 seleniumserver.stop();  
} }