**TestNG Listeners & Extent Reports in Selenium**

In this post, we see TestNG listeners. Listeners “listen” to the event defined in the selenium script and behave accordingly. The main purpose of using listeners is to create logs. There are many types of listeners such as WebDriver Listeners and TestNG Listeners.  
  
Here in this post, we see TestNG Listeners. Using TestNG listeners we could generate logs and customize TestNG Reports.

Let’s see how to implement **TestNG Listeners.**

**Step 1: Create a Class Listeners extends TestListenerAdapter class**

import org.testng.ITestResult;  
import org.testng.TestListenerAdapter;  
  
public class Listeners extends TestListenerAdapter {  
 public void onTestStart(ITestResult tr) {  
 System.out.println("test is started");  
 }  
  
 public void onTestSuccess(ITestResult tr) {  
 System.out.println(" test is passed");  
 }  
  
 public void onTestFailure(ITestResult tr) {  
 System.out.println(" test is failed");  
 }  
  
 public void onTestSkipped(ITestResult tr) {  
 System.out.println(" test is skipped");  
 }  
  
}

**Step 2: Create a Test Case LoginTest.java**  
  
import org.testng.Assert;  
import org.testng.annotations.Test;  
  
public class LoginTest {  
  
 @Test  
 void setup()  
 {  
 Assert.fail();  
 }  
   
 @Test  
 void loginByEmail()  
 {  
 Assert.assertTrue(true);  
 }  
   
 @Test(dependsOnMethods={"setup"})  
 void loginByFacebook()  
 {  
 Assert.assertTrue(true);  
 }  
   
}

**Step 3: Create listener.xml file to run test case**

**ExtentReprts**

Third party API which will generate attractive reports for test cases.

**Pre-requisites to generate extent reports**

1) Download extent report jars & xml

URL:http://extentreports.com/community/#

2) Unzip file--> xml file along with jar files

3) Add all jars to project build path

4) copy extent-config.xml file into project home directory.

**Listener.java**

import java.io.IOException;

import org.testng.ITestContext;

import org.testng.ITestResult;

import org.testng.TestListenerAdapter;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

import com.aventstack.extentreports.Status;

import com.aventstack.extentreports.markuputils.ExtentColor;

import com.aventstack.extentreports.markuputils.MarkupHelper;

import com.aventstack.extentreports.reporter.ExtentHtmlReporter;

import com.aventstack.extentreports.reporter.configuration.ChartLocation;

import com.aventstack.extentreports.reporter.configuration.Theme;

public class Listeners extends TestListenerAdapter

{

public ExtentHtmlReporter htmlReporter;

public ExtentReports extent;

public ExtentTest logger;

public void onStart(ITestContext testContext)

{

htmlReporter=new ExtentHtmlReporter(System.getProperty("user.dir")+ "/test-output/myReport.html");//specify location of the report

htmlReporter.loadXMLConfig(System.getProperty("user.dir")+ "/extent-config.xml");

extent=new ExtentReports();

extent.attachReporter(htmlReporter);

extent.setSystemInfo("Host name","localhost");

extent.setSystemInfo("Environemnt","QA");

extent.setSystemInfo("user","pavan");

htmlReporter.config().setDocumentTitle("Automation Report"); // Tile of report

htmlReporter.config().setReportName("Functional Testing"); // name of the report

htmlReporter.config().setTestViewChartLocation(ChartLocation.TOP); //location of the chart

htmlReporter.config().setTheme(Theme.STANDARD);

}

public void onTestSuccess(ITestResult tr)

{

logger=extent.createTest(tr.getName()); // create new entry in th report

logger.log(Status.PASS,MarkupHelper.createLabel(tr.getName(),ExtentColor.GREEN)); // send the passed information to the report with GREEN color highlighted

}

public void onTestFailure(ITestResult tr)

{

logger=extent.createTest(tr.getName()); // create new entry in th report

logger.log(Status.FAIL,MarkupHelper.createLabel(tr.getName(),ExtentColor.RED)); // send the passed information to the report with GREEN color highlighted

String screenshotPath=System.getProperty("user.dir")+"\\Screenshots\\"+tr.getName()+".png";

try {

logger.fail("Screenshot is below:" + logger.addScreenCaptureFromPath(screenshotPath));

} catch (IOException e) {

e.printStackTrace();

}

}

public void onTestSkipped(ITestResult tr)

{

logger=extent.createTest(tr.getName()); // create new entry in th report

logger.log(Status.SKIP,MarkupHelper.createLabel(tr.getName(),ExtentColor.ORANGE));

}

public void onFinish(ITestContext testContext)

{

extent.flush();

}

}

**LoginTest.Java**

import java.io.File;

import java.io.IOException;

import org.openqa.selenium.By;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.testng.Assert;

import org.testng.ITestResult;

import org.testng.annotations.AfterClass;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.Test;

import org.apache.commons.io.FileUtils;

public class LoginTest {

WebDriver driver;

@Test(priority=1)

void loginTest() throws InterruptedException

{

System.setProperty("webdriver.chrome.driver", "C://Drivers//chromedriver\_win32//chromedriver.exe");

driver = new ChromeDriver();

driver.get("http://opensource.demo.orangehrmlive.com");

driver.findElement(By.name("txtUsername")).sendKeys("Admin");

driver.findElement(By.name("txtPassword")).sendKeys("admin");

driver.findElement(By.name("Submit")).click();

Thread.sleep(2000);

Assert.assertEquals(driver.getTitle(), "OrangeHRM");

}

@Test(priority=2)

void checkTotalNoOfEmployees()

{

Assert.assertTrue(true);

}

@Test(priority=3,dependsOnMethods={"loginTest"})

void checkNoOfEmployeesEnabled()

{

Assert.assertTrue(true);

}

@AfterMethod

public void captureScreen(ITestResult result) throws IOException

{

if(result.getStatus()==ITestResult.FAILURE)

{

TakesScreenshot ts=(TakesScreenshot)driver;

File source=ts.getScreenshotAs(OutputType.FILE); // capture screenshot file

File target=new File(System.getProperty("user.dir")+"/Screenshots/"+result.getName()+".png");

FileUtils.copyFile(source,target);

System.out.println("screenshot captured");

}

}

@AfterClass

void closeBrowser()

{

driver.quit();

}

}

**listener.xml**