Test Ng Listeners

Test ng provides list of listeners as part of framework

* IAnnotationTransformer ([doc](https://testng.org/doc/documentation-main.html#annotationtransformers), [javadoc](https://javadoc.io/doc/org.testng/testng/latest/org/testng/IAnnotationTransformer.html))
* IAnnotationTransformer2 ([doc](https://testng.org/doc/documentation-main.html#annotationtransformers), [javadoc](https://javadoc.io/doc/org.testng/testng/latest/org/testng/IAnnotationTransformer2.html))
* IHookable ([doc](https://testng.org/doc/documentation-main.html#ihookable), [javadoc](https://javadoc.io/doc/org.testng/testng/latest/org/testng/IHookable.html))
* IInvokedMethodListener (doc, [javadoc](https://javadoc.io/doc/org.testng/testng/latest/org/testng/IInvokedMethodListener.html))
* IMethodInterceptor ([doc](https://testng.org/doc/documentation-main.html#methodinterceptors), [javadoc](https://javadoc.io/doc/org.testng/testng/latest/org/testng/IMethodInterceptor.html))
* IReporter ([doc](https://testng.org/doc/documentation-main.html#logging-reporters), [javadoc](https://javadoc.io/doc/org.testng/testng/latest/org/testng/IReporter.html))
* ISuiteListener (doc, [javadoc](https://javadoc.io/doc/org.testng/testng/latest/org/testng/ISuiteListener.html))
* ITestListener ([doc](https://testng.org/doc/documentation-main.html#logging-listeners), [javadoc](https://javadoc.io/doc/org.testng/testng/latest/org/testng/ITestListener.html))

Listners is a piece fo code that will listen when an event accurs

|  |  |
| --- | --- |
| void | [**onFinish**](https://javadoc.io/static/org.testng/testng/6.13/org/testng/ITestListener.html#onFinish-org.testng.ITestContext-)(**[ITestContext](https://javadoc.io/static/org.testng/testng/6.13/org/testng/ITestContext.html" \o "interface in org.testng)** context)  Invoked after all the tests have run and all their Configuration methods have been called. |
| void | [**onStart**](https://javadoc.io/static/org.testng/testng/6.13/org/testng/ITestListener.html#onStart-org.testng.ITestContext-)(**[ITestContext](https://javadoc.io/static/org.testng/testng/6.13/org/testng/ITestContext.html" \o "interface in org.testng)** context)  Invoked after the test class is instantiated and before any configuration method is called. |
| void | [**onTestFailedButWithinSuccessPercentage**](https://javadoc.io/static/org.testng/testng/6.13/org/testng/ITestListener.html#onTestFailedButWithinSuccessPercentage-org.testng.ITestResult-)(**[ITestResult](https://javadoc.io/static/org.testng/testng/6.13/org/testng/ITestResult.html" \o "interface in org.testng)** result)  Invoked each time a method fails but has been annotated with successPercentage and this failure still keeps it within the success percentage requested. |
| void | [**onTestFailure**](https://javadoc.io/static/org.testng/testng/6.13/org/testng/ITestListener.html#onTestFailure-org.testng.ITestResult-)(**[ITestResult](https://javadoc.io/static/org.testng/testng/6.13/org/testng/ITestResult.html" \o "interface in org.testng)** result)  Invoked each time a test fails. |
| void | [**onTestSkipped**](https://javadoc.io/static/org.testng/testng/6.13/org/testng/ITestListener.html#onTestSkipped-org.testng.ITestResult-)(**[ITestResult](https://javadoc.io/static/org.testng/testng/6.13/org/testng/ITestResult.html" \o "interface in org.testng)** result)  Invoked each time a test is skipped. |
| void | [**onTestStart**](https://javadoc.io/static/org.testng/testng/6.13/org/testng/ITestListener.html#onTestStart-org.testng.ITestResult-)(**[ITestResult](https://javadoc.io/static/org.testng/testng/6.13/org/testng/ITestResult.html" \o "interface in org.testng)** result)  Invoked each time before a test will be invoked. |
| void | [**onTestSuccess**](https://javadoc.io/static/org.testng/testng/6.13/org/testng/ITestListener.html#onTestSuccess-org.testng.ITestResult-)(**[ITestResult](https://javadoc.io/static/org.testng/testng/6.13/org/testng/ITestResult.html" \o "interface in org.testng)** result)  Invoked each time a test succeeds. |

TestListners.java

package test;

import org.testng.ITestContext;

import org.testng.ITestListener;

import org.testng.ITestResult;

public class TestListners implements ITestListener{

public void onStart(ITestContext context)

{ //Invoked after the test class is instantiated and before any configuration method is called.

System.out.println("onstart method started");

}

public void onTestStart(ITestResult result)

{ //Invoked each time before a test will be invoked.

System.out.println("New Test started :"+ result.getName());

}

public void onTestSuccess(ITestResult result)

{ //Invoked each time a test succeeds.

System.out.println("on Test sucess method:"+ result.getName());

}

public void onTestFailure(ITestResult result)

{ //Invoked each time a test fails.

System.out.println("on Test failure method:"+ result.getName());

}

public void onTestSkipped(ITestResult result)

{ //Invoked each time a test is skipped.

System.out.println("on Test failure method:"+ result.getName());

}

public void onTestFailedButWithinSuccessPercentage(ITestResult result)

{ /\*

Invoked each time a method fails but has been annotated with

successPercentage and this failure

still keeps it within the success percentage requested

\*/

System.out.println("onTest Failed Bu tWithin Success Percentage:"+ result.getName());

}

}

ListenersDemo.java

**package** test;

**import** org.openqa.selenium.PageLoadStrategy;

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

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

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

**import** org.openqa.selenium.edge.EdgeDriver;

**import** org.testng.Assert;

**import** org.testng.SkipException;

**import** org.testng.annotations.AfterTest;

**import** org.testng.annotations.BeforeTest;

**import** org.testng.annotations.Listeners;

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

@Listeners(TestListners.**class**)

**public** **class** ListenersDemo {

ChromeOptions chromeOptions;

WebDriver driver;

String appurl;

@BeforeTest

**public** **void** LaunchBrowser() {

System.***out***.println("Launch browser");

System.*setProperty*("webdriver.chrome.driver", "C:\\SeleniumDrivers\\chromedriver\\chromedriver.exe");

chromeOptions = **new** ChromeOptions();

chromeOptions.setPageLoadStrategy(PageLoadStrategy.***EAGER***);

chromeOptions.addArguments("--remote-allow-origins=\*");

driver = **new** ChromeDriver(chromeOptions);

appurl = "https://adactinhotelapp.com/";

}

@AfterTest

**public** **void** CloseBrowser() {

System.***out***.println("Close browser");

driver.close();

}

@Test

**public** **void** VerifyTitle() {

driver.get(appurl);

String expectedTitle = "Adactin.com - Hotel Reservation System";

String actualTitle = driver.getTitle();

Assert.*assertEquals*(expectedTitle, actualTitle, "Title are not matching");

}

@Test

**public** **void** VerifyLogin() {

System.***out***.println("Login test called");

}

@Test

**public** **void** SkipTest() {

System.***out***.println("Skip test called");

**throw** **new** SkipException("Skipping the test ");

}

@Test

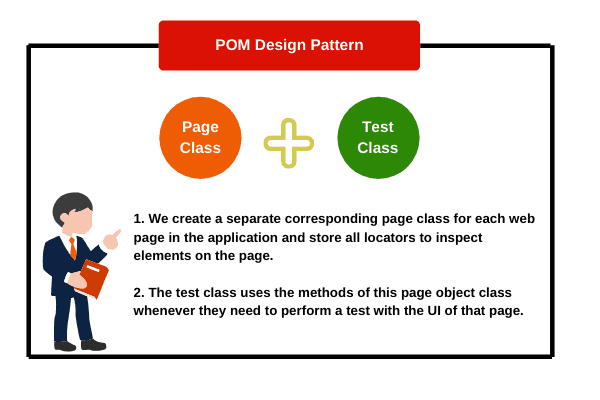
**public** **void** FailTest() {

System.***out***.println("Fail test called");

Assert.*assertEquals*(**false**, **true**);

}

}



**package** test;

**import** org.openqa.selenium.By;

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

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

**public** **class** HotelLoginPage {

// Create a web driver reference variable

WebDriver driver;

**public** HotelLoginPage(WebDriver driver)

{

**this**.driver=driver;

}

// identify all web elements

By username=By.*xpath*("//input[@id='username']");

By password=By.*xpath*("//input[@id='password']") ;

By btnLogin=By.*xpath*("//input[@id='login']") ;

**public** **void** setUsername(String strUsername )

{

driver.findElement(username).sendKeys(strUsername);

}

**public** **void** setPassword(String strPassword )

{

driver.findElement(password).sendKeys(strPassword);

}

**public** **void** clickLogin()

{

driver.findElement(btnLogin).click();

}

**public** String titleText()

{

String strtitle=driver.getTitle();

**return** strtitle;

}

**public** **void** doLogin(String strUsername,String strPassword)

{

**this**.setUsername(strUsername);

**this**.setPassword(strPassword);

**this**.clickLogin();

}

}

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

driver.get(appurl);

HotelLoginPage lp= **new** HotelLoginPage(driver);

lp.doLogin("vtestuser02","selenium");

String expectedTitle = "Adactin.com - Search Hotel";

String actualTitle = lp.titleText();

Assert.*assertEquals*(expectedTitle, actualTitle, "Title are not matching");

}