

# Using Listeners

In this topic, we will see the list of TestNG Listeners.

## WE'LL COVER THE FOLLOWING ^

- TestNG listeners
- Declaring listeners
  - In testng.xml
  - In TestClass

## TestNG listeners #

There are several interfaces that allow us to change the behavior of TestNG at runtime:

- **IAlterSuiteListener** - on implementing this interface, we have method **alter** to alter the test suite
- **IAnnotationTransformer** - on implementing this interface, we can alter **@Test**, **@DataProvider**, **@Factory**, **@Listener** annotations
- **IClassListener** - on implementing this interface, we can call **onBeforeClass** and **onAfterClass**
- **IConfigurable** - on implementing the interface, **run** method will be invoked instead of each configuration method found
- **IConfigurationListener** - on implementing this interface, we can call methods **onConfigurationSuccess**, **onConfigurationFailure**, **onConfigurationSkip** and **beforeConfiguration**
- **IDataProviderListener** - method **beforeDataProviderExecution** and **afterDataProviderExecution**
- **IExecutionListener** - **onExecutionStart** and **onExecutionFinish**
- **IHookable** - this method has a method **run** and on implementing, **run** method will be called instead of **@Test** method

- **IInvokedMethodListener** - on implementing this interface, we have method `beforeInvocation`, `afterInvocation`, `beforeInvocation`, `afterInvocation` that gets invoked before and after a method is invoked by TestNG, irrespective of whether they pass/fail or gets skipped
- **IMethodInterceptor** - on implementing the interface, we can call *intercept* to alter the list of test methods that TestNG is about to run
- **IReporter** - on implementing the interface, we have method `generateReport` to create custom reports
- **ISuiteListener** - on implementing the interface, we can call `onStart` and `onFinish`
- **ITestListener** - on implementing the interface, we have methods `onTestStart`, `onTestSuccess`, `onTestFailure`, `onTestSkipped`, `onTestFailedButWithinSuccessPercentage`, `onTestFailedWithTimeout`, `onStart`, `onFinish`

## Declaring listeners #

In testng.xml #

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Sample Test Suite" parallel="tests" thread-count="5">
  <listeners>
    <listener class-name="com.example.SampleTestListener" />
  </listeners>
  <test name="Sample Test">
    ...
  </test>
</suite>
```

Listeners can be configured for the entire test suite in *testng.xml* like above. Under `<listeners>`, multiple `<listener>` can be added.

In TestClass #

```
@Listeners( { com.example.SampleTestListener.class } )
public class TestClass {
  ....
}
```

Listeners can be configured only for certain test classes like the above code snippet by adding `@Listeners({ ListenerA.class, ListenerB.class, ...})` at the class level.

---

That is how listeners work with TestNG. In the next lesson, you will study how to restart a failed test.