# Test Framework Overview


| Description | | TestNG (6.10) | Junit (Junit 5) |
|---|---|---|---|
| Prerequisite | Execute Before / After each suite | @BeforeSuite<br>@AfterSuite | **Not Available** |
| | Execute Before / After each class | @BeforeClass<br>@AfterClass | @BeforeAll<br>@AfterAll |
| | | | Executed Before/ After all @Test, @RepeatedTest, @ParameterizedTest, and @TestFactory methods in the current class; analogous to JUnit 4's @BeforeClass. |
| | Execute Before / After each Method | @BeforeMethod<br>@AfterMethod | **Not Available** |
| | Execute Before / After each Test | @BeforeTest<br>@AfterTest | @BeforeEach<br>@AfterEach |
| | | | Executed before each @Test, @RepeatedTest, @ParameterizedTest, or @TestFactory method in the current class; analogous to JUnit 4's @Before |
| | Execute Before / After each group | @BeforeGroups<br>@AfterGroups | **Not Available** |
| Test Annotation | Test Method | @Test | @Test |
| | Description | @Test(description='Login Test') | @Test<br>@*DisplayName*("Custom test name containing spaces") |
| | | | ***Inherit the '*DisplayNameGenerator.ReplaceUnderscores*' class to update the report format |
| | Sequential Execution | @Test(priority=1) | @Test<br>@ Order(1) |
| | Execute one test after based previous result | @Test(dependsOnMethods = 'methodName') | **Not Available** |
| | Grouping the Test | @Test(groups='Regression') | @Tag("Regression") |
| | Ignore Test | @Test (enabled =false) | @Test |
| | | | @*Disabled*('Execute after fix') |
| | | | @*EnabledOnOs*({ LINUX, MAC })<br>@*DisabledOnOs*(WINDOWS) |
| | | | @*EnabledIfSystemProperty*(named = "os.arch", matches="true")<br>@*DisabledIfSystemProperty*(named = "ci-server", matches = "true") |

# Test Framework Overview


| Description | | TestNG (6.10) | Junit (Junit 5) |
|---|---|---|---|
| | | | *@EnabledIfEnvironmentVariable*(named = "ENV", matches = "staging-server")<br>*@DisabledIfEnvironmentVariable*(named = "ENV", matches = ".*development.*") |
| | | | *@EnabledIf*("2 * 3 == 6")<br>*@DisabledIf*("2 * 3 == 6") |
| | Time Out | @Test (timeout = 100) | **Not Available** |
| | Exception | @Test(expectedExceptions = ArithmeticExpeption.class) | **Not Available** |
| | Execute the same test at multiple times | @Test(InvocationCount=10 ,threadPollSize=5, invocationTimeOut=10, successPercentage=50)<br>(timeout in ms)<br>(percentage of success expected from this method) | **@RepeatedTest(10)**<br>**@RepeatedTest**(value = 1, name = "{displayName} {currentRepetition}/{totalRepetitions}")<br>**@DisplayName("Repeat!")** |
| | Always runs even if the parameters on which the method depends fails. | @Test(alwaysRun = true) | **Not Available** |
| | Class level Annotation | **@Test(groups = { "checkin-test" })**<br>public class All {<br>*@Test(groups = { "func-test" )*<br>*public void method1() { ... }*<br>*public void method2() { ... }*<br>}<br>***Method – part of both group, method2 is part of 'checkin-test'*<br>*** Method 1 and 2 are Test Methods* | **@Disabled("Disabled until this bug#11 fixed")**<br>**@Tag("Disabled")**<br>class DisabledClassDemo {<br>@Test<br>Public void testWillBeSkipped() {<br>}<br>} |
| **Parameterizatio** | **TestNG - [@DataProvider]** Supply the test data to test case using @dataprovider annotation, | **@DataProvider(name = "test1")**<br>*public static Object[][] validUserName() {*<br>*Read the value from excel / DB /othes and storied in the double dimension object*<br>} | **@ParameterizedTest**<br>***@ValueSource***(strings = { "racecar", "radar", "able was I ere I saw elba" })<br>*@ValueSource*(ints = { 1, 2, 3 })<br>*@ValueSource*(strings = "SECONDS") |

# Test Framework Overview


| Description | | TestNG (6.10) | Junit (Junit 5) |
|---|---|---|---|
| | To execute same test in multiple times with different data. | **@Test(dataProvider = "test1", dataProviderClass = '*validUserName* .class ')**<br>public void verifyValidUserName(String UserName) {<br>} | **@*EnumSource***(value = TimeUnit.class, names = { "DAYS", "HOURS" })<br>*@MethodSource*("stringProvider")<br>**@*CsvSource***({ "apple, banana" , "Orange, Mango" })<br>**@*CsvFileSource*(**resources = "/two-column.csv", numLinesToSkip = 1) |
| | **Test NG - [@Parameters]** Supply the test data from TestNG.xml to test case using @parameters annotation | **@Parameters("Name")**<br>@Test<br>public void verifyValidUserName(String UserName) {<br>}<br>**TestNG Suite File (xml)**<br><suite name = "Suite1"><br><test name = "test1"><br>**<parameter name = "Name" value="user01"/>**<br><classes><br><class name = "ParameterizedTest1" /><br></classes><br></test><br></suite> | |
| **Assertion** | Assertion to validate the actual with expected condition. | Assert.assertEquals(actual,expected);<br>Assert.assertNotEquals(actual,expected,Message);<br>Assert.assertTrue(condition);<br>Assert.assertFalse(condition);<br>Assert.assertNull(object);<br>Assert.assertNotNull(object); | **import static org.junit.jupiter.api.Assertions.assertEquals;**<br>assertEquals(actual, expected);<br>assertNotEquals(actual, expected);<br>assertEquals(actual, expected, Message);<br>assertTrue(condition, message);<br>assertFalse(condition, message);<br>assertAll("last name",<br>() -> assertTrue(lastName.startsWith("D")),<br>() -> assertTrue(lastName.endsWith("e")) );<br>Assert.assertNull(object);<br>Assert.assertNotNull(object);<br>assertTimeout(ofMillis(10), () -> {<br>// Simulate task that takes more than 10 ms.<br>Thread.sleep(100); }); |
| E | | <suite name="My test suite"> | **@RunWith(JUnitPlatform.class)** |

# Test Framework Overview


| Description | | TestNG (6.10) | Junit (Junit 5) |
|---|---|---|---|
| | Execute the automation test suite using test framework. | <test name=”testing”><br><classes><br><class name=”com.fsecure.demo.testng.Test1”/><br><class name=”com.fsecure.demo.testng.Test2”/><br></classes><br></test><br></suite> | @SelectPackages({"com.howtodoinjava.junit5.examples.packageA","com.howtodoinjava.junit5.examples.packageB"})<br>public class JUnit5TestSuiteExample<br>{<br>} |
| | | <class name="com.easy.entry.AddTestCase"><br><methods><br><include name="addLocTestCase" /><br><exclude name="addEmplTestCase" /><br></methods><br></class> | **Single / Multiple Class**<br>@SelectClasses( { ClassATest.class, ClassBTest.class, ClassCTest.class } ) |
| | | <test name="Simple example"><br><groups><br><run><br><include name="checkintest"/><br><exclude name="broken"/><br></run><br></groups> | **Include / Exclude Package**<br>@SelectPackages("com.howtodoinjava.junit5.examples")<br>@IncludePackages("com.howtodoinjava.junit5.examples.packageC")<br>@ExcludePackages("com.howtodoinjava.junit5.examples.packageC") |
| | | | **Include / Exclude class**<br>@IncludeClassNamePatterns({"^.*ATests?$"})<br>@ExcludeClassNamePatterns({"^.*ATests?$"}) |
| | | | **Include/ Exclude Tags**<br>@IncludeTags("production")<br>@ExcludeTags("development") |
| | | | **JUNIT 4**<br>@RunWith(Suite.class)<br>@Suite.SuiteClasses({<br>JunitTest1.class,<br>JunitTest2.class<br>})<br>public class JunitTest5 {<br>} |
| **Parall** | Execute Tests in parallel | ***Suite Level Parallel Execution***<br>cmd >> java org.testng.TestNG -suitethreadpoolsize 3 testng1.xml testng2.xml testng3.xml | **Resources -junit-platform.properties file**<br>junit.jupiter.execution.parallel.enabled=true<br>junit.jupiter.execution.parallel.config.strategy=dynamic |

# Test Framework Overview

JUnit VS TestNG

| Description | | TestNG (6.10) | Junit (Junit 5) |
|---|---|---|---|
| | | ***Method Level Parallel Execution***<br><suite name="My suite" parallel="methods" thread-count="5"> | |
| | | ***Tests Level Parallel Execution***<br><suite name="My suite" parallel="tests" thread-count="5"> | **@Execution(ExecutionMode.CONCURRENT)**<br><br>*@Execution(ExecutionMode.CONCURRENT)*<br>*public class LoginTest extends BaseTest {*<br>*@Test*<br>*public void invalidLoginTest_(){ }*<br>*@Test*<br>*public void invalidLoginTest_(){ }*<br>*}*<br>***SAME_THREAD***<br>*Force execution in the same thread used by the parent. For example, when used on a test method, the test method will be executed in the same thread as any @BeforeAll or @AfterAll methods of the containing test class.*<br><br>***CONCURRENT***<br>*Execute concurrently unless a resource lock forces execution in the same thread.* |
| | | ***Class Level Parallel Execution***<br><suite name="My suite" parallel="classes" thread-count="5"> | |
| | | ***Instances Level Parallel Execution***<br><suite name="My suite" parallel="instances" thread-count="5"> | |
| **Listeners** | | IAnnotationTransformer | **Listener** |
| | | IAnnotationTransformer2 | **TestExecutionListener**<br>SummaryGeneratingListener<br>LegacyXmlReportGeneratingListener |
| | | IHookable | |
| | | IInvokedMethodListener | |
| | | IMethodInterceptor | |
| | | IReporter | |
| | | ISuiteListener | |
| | | ITestListener | |
| | | <suite> | **Extension Model** |

# Test Framework Overview


| Description | | TestNG (6.10) | Junit (Junit 5) |
|---|---|---|---|
| | | <listeners><br><listener class-name="com.example.MyListener" /><br><listener class-name="com.example.MyMethodInterceptor" /><br></listeners> | **@ExtendWith(TestLifeCycleExtensions.class)**<br>Public class TestInstancePostProcessor, BeforeAllCallback, BeforeEachCallback, BeforeTestExecutionCallback, AfterTestExecutionCallback, AfterEachCallback, AfterAllCallback{<br>//Override & modify as per our need<br>}<br><br>@ExtendWith(TestLifeCycleExtensions.class)<br>public class test1 {<br>} |