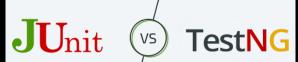
JUnit (vs) TestNG



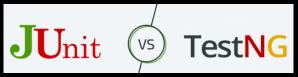
Description		TestNG (6.10)	Junit (Junit 5)
	Execute Before /	@BeforeSuite	Not Available
	After each suite	@AfterSuite	
	Execute Before /	@BeforeClass	@BeforeAll
	After each class	@AfterClass	@AfterAll
			Executed Before/ After all @Test,
			@RepeatedTest, @ParameterizedTest, and
te te			@TestFactory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





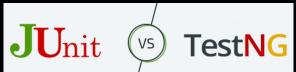
	rest transcover & verview				
Desc	cription	TestNG (6.10)	Junit (Junit 5)		
			<pre>@EnabledIfEnvironmentVariable(named = "ENV", matches = "staging-server") @DisabledIfEnvironmentVariable(named = "ENV", matches = ".*development.*")</pre>		
			@EnabledIf("2 * 3 == 6")		
		V. 5	@DisabledIf("2 * 3 == 6")		
	T: 0 :	OT 1/11 1 100)			
	Time Out	@Test (timeout = 100)	Not Available		
	Exception	@Test(expectedExceptions =	Not Available		
		ArithmeticExpeption.class)			
	Execute the same	@Test(InvocationCount=10 ,threadPollSize=5,	@RepeatedTest(10)		
	test at multiple times	invocationTimeOut=10, successPercentage=50) (timeout in ms)	@RepeatedTest(value = 1, name =		
	tilles	(percentage of success expected from this	"{displayName}		
		method)	{currentRepetition}/{totalRepetitions}")		
		The chief	@DisplayName("Repeat!")		
	Always runs even if the parameters on which the method depends fails.	@Test(alwaysRun = true)	Not Available		
	Class level Annotation	<pre>@Test(groups = { "checkin-test" })</pre>	<pre>@Disabled("Disabled until this bug#11 fixed") @Tag("Disabled")</pre>		
		public class All {	class Disabled / class Disabled Class Demo {		
		@Test(groups = { "func-test")	@Test		
		public void method1() { }	Public void testWillBeSkipped() {		
		public void method2() { }	}		
		}	}		
		**Method – part of both group, method2 is			
		part of 'checkin-test'			
		** Method 1 and 2 are Test Methods			
	TestNG - [@DataProvider] Supply the test	@DataProvider(name = "test1")	@ParameterizedTest		
Parameterizatio					
		mublic static Object[[[] walld last Alama () (@ValueSource(strings = { "racecar", "radar", "able was I ere I saw elba" })		
ete	data to test case	<pre>public static Object[][] validUserName() { Read the value from excel / DB /othes and</pre>	**		
am	using	storied in the double dimension object	@ValueSource(ints = { 1, 2, 3 })		
Par	@dataprovider	}	@ValueSource(strings = "SECONDS")		
	annotation,] *			





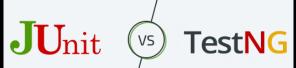
Desc	cription	TestNG (6.10)	Junit (Junit 5)
	To execute same test in multiple times with	<pre>@Test(dataProvider = "test1", dataProviderClass = 'validUserName .class ') public void verifyValidUserName(String</pre>	<pre>@EnumSource(value = TimeUnit.class, names = { "DAYS", "HOURS" })</pre>
	different data.	UserName) { }	@MethodSource("stringProvider")
			@CsvSource({ "apple, banana", "Orange,
	Test NG -	@Parameters("Name")	Mango" })
	[@Parameters]	@Test	
Just 3	Supply the test	public void verifyValidUserName(String	@CsvFileSource(resources = "/two-
	data from TestNG.xml to test	UserName) {	column.csv", numLinesToSkip = 1)
	case using		
	@parameters	TestNG Suite File (xml)	
\	annotation	<pre><suite name="Suite1"></suite></pre>	
		<test name="test1"></test>	
		<pre><parameter name="Name" value="user01"></parameter></pre>	
		<classes></classes>	
		<class name="ParameterizedTest1"></class>	
V.			
	Assertion to	Assert.assertEquals(actual,expected);	import static org.junit.jupiter.api.Assertions.
	validate the actual	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	assertEquals;
	with expected		assertEquals(actual, expected);
	condition.	Assert.assertNotEquals(actual,expected,Messa	assertNotEquals(actual, expected);
		ge);	assertEquals(actual, expected, Message);
=		Assert.assertTrue(condition);	assertTrue(condition, message);
ssertion		Assert.assertFalse(condition);	assertFalse(condition, message);
ser		Assert.assertNull(object);	assertAll("last name",
As		Assert.assertNotNull(object);	<pre>() -> assertTrue(lastName.startsWith("D")), () -> assertTrue(lastName.endsWith("e")));</pre>
			Assert.assertNull(object);
			Assert.assertNotNull(object);
			assertTimeout(ofMillis(10), () -> {
			// Simulate task that takes more than 10 ms.
			Thread.sleep(100); });
Е		<suite name="My test suite"></suite>	@RunWith(JUnitPlatform.class)





Desc	ription	TestNG (6.10)	Junit (Junit 5)
	Execute the automation test suite using test framework.	<test name="testing"> <classes> <class name="com.fsecure.demo.testng.Test1"></class> <class name="com.fsecure.demo.testng.Test2"></class> </classes> </test>	@SelectPackages({"com.howtodoinjava.juni 5.e xamples.packageA","com.howtodoinjava.jun t5.examples.packageB"}) public class JUnit5TestSuiteExample { } Single / Multiple Class
			@SelectClasses({ ClassATest.class, ClassBTest.class, ClassCTest.class })
		<pre>name="com.easy.entry.AddTestCase"></pre>	Include / Exclude Package @SelectPackages("com.howtodoinjava.junit5 .examples") @IncludePackages("com.howtodoinjava.junit 5.examples.packageC") @ExcludePackages("com.howtodoinjava.junit 5.examples.packageC") Include / Exclude class @IncludeClassNamePatterns({"^.*ATests?\$"}) @ExcludeClassNamePatterns({"^.*ATests?\$"}) Include / Exclude Tags @IncludeTags("production") @ExcludeTags("development")
			JUNIT 4
			<pre>@RunWith(Suite.class) @Suite.SuiteClasses({ JunitTest1.class, JunitTest2.class }) public class JunitTest5 { }</pre>
Parall	Execute Tests in parallel	Suite Level Parallel Execution cmd >> java org.testng.TestNG - suitethreadpoolsize 3 testng1.xml testng2.xml testng3.xml	Resources -junit-platform.properties file junit.jupiter.execution.parallel.enabled=true junit.jupiter.execution.parallel.config.strategy =dynamic





Description	TestNG (6.10)	Junit (Junit 5)
	Method Level Parallel Execution <suite name="My suite" parallel="methods" thread-count="5"></suite>	@Execution(ExecutionMode.CONCURRE NT)
	Tests Level Parallel Execution <suite name="My suite" parallel="tests" thread-count="5"> Class Level Parallel Execution <suite name="My suite" parallel="classes" thread-count="5"> Instances Level Parallel Execution <suite name="My suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="My suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="My suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="My suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="My suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="My suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="My suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="My suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="My suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="My suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="my suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="my suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="my suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="my suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="my suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="my suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution <suite name="my suite" parallel="instances" thread-count="5"> Instances Level Parallel Execution Instances Level Parallel Execution Instances Level Parallel Execution Instances Level Parallel Execution Instances Level Parallel Execution</suite></suite></suite></suite></suite></suite></suite></suite></suite></suite></suite></suite></suite></suite></suite></suite></suite></suite></suite>	@Execution(ExecutionMode.CONCURRENT) public class LoginTest extends BaseTest { @Test public void invalidLoginTest_(){ } @Test public void invalidLoginTest_(){ } } SAME_THREAD 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. CONCURRENT Execute concurrently unless a resource lock forces execution in the same thread.
Listeners	IAnnotationTransformer IAnnotationTransformer2 IHookable IInvokedMethodListener IMethodInterceptor IReporter ISuiteListener ITestListener <suite></suite>	Listener TestExecutionListener SummaryGeneratingListener LegacyXmlReportGeneratingListener Extension Model



JUnit (vs) TestNG

Des	cription	TestNG (6.10)	Junit (Junit 5)
		<pre>teners> listener class- name="com.example.MyListener" /> </pre>	@ExtendWith(TestLifeCycleExtensions.class) Public class TestInstancePostProcessor, BeforeAllCallback, BeforeEachCallback, BeforeTestExecutionCallback, AfterTestExecutionCallback, AfterEachCallback, AfterAllCallback{ //Override & modify as per our need }
\(\sigma\)			<pre>@ExtendWith(TestLifeCycleExtensions.class) public class test1 { }</pre>

