

JUnit 5

JUnit5 Annotations

Annotation	Description
@Test	Denotes that a method is a test method. Unlike JUnit 4's @Test annotation, this annotation does not declare any attributes, since test extensions in JUnit Jupiter operate based on their own dedicated annotations. Such methods are <i>inherited</i> unless they are <i>overridden</i> .
@ParameterizedTest	Denotes that a method is a parameterized test . Such methods are <i>inherited</i> unless they are <i>overridden</i> .
@RepeatedTest	Denotes that a method is a test template for a repeated test . Such methods are <i>inherited</i> unless they are <i>overridden</i> .
@TestFactory	Denotes that a method is a test factory for dynamic tests . Such methods are <i>inherited</i> unless they are <i>overridden</i> .
@TestTemplate	Denotes that a method is a template for test cases designed to be invoked multiple times depending on the number of invocation contexts returned by the registered providers . Such methods are <i>inherited</i> unless they are <i>overridden</i> .
@TestMethodOrder	Used to configure the test method execution order for the annotated test class; similar to JUnit 4's @FixMethodOrder. Such annotations are <i>inherited</i> .
@TestInstance	Used to configure the test instance lifecycle for the annotated test class. Such annotations are <i>inherited</i> .
@DisplayName	Declares a custom display name for the test class or test method. Such annotations are not <i>inherited</i> .
@DisplayNameGeneration	Declares a custom display name generator for the test class. Such annotations are <i>inherited</i> .

Annotation	Description
@BeforeEach	Denotes that the annotated method should be executed <i>before</i> each @Test, @RepeatedTest, @ParameterizedTest, or @TestFactory method in the current class; analogous to JUnit 4's @Before. Such methods are <i>inherited</i> unless they are <i>overridden</i> .
@AfterEach	Denotes that the annotated method should be executed <i>after</i> each @Test, @RepeatedTest, @ParameterizedTest, or @TestFactory method in the current class; analogous to JUnit 4's @After. Such methods are <i>inherited</i> unless they are <i>overridden</i> .
@BeforeAll	Denotes that the annotated method should be executed <i>before</i> all @Test, @RepeatedTest, @ParameterizedTest, and @TestFactory methods in the current class; analogous to JUnit 4's @BeforeClass. Such methods are <i>inherited</i> (unless they are <i>hidden</i> or <i>overridden</i>) and must be static (unless the "per-class" test instance lifecycle is used).
@AfterAll	Denotes that the annotated method should be executed <i>after</i> all @Test, @RepeatedTest, @ParameterizedTest, and @TestFactory methods in the current class; analogous to JUnit 4's @AfterClass. Such methods are <i>inherited</i> (unless they are <i>hidden</i> or <i>overridden</i>) and must be static (unless the "per-class" test instance lifecycle is used).
@Nested	Denotes that the annotated class is a non-static nested test class . @BeforeAll and @AfterAll methods cannot be used directly in a @Nested test class unless the "per-class" test instance lifecycle is used. Such annotations are not <i>inherited</i> .
@Tag	Used to declare tags for filtering tests , either at the class or method level; analogous to test groups in

Annotation	Description
	TestNG or Categories in JUnit 4. Such annotations are <i>inherited</i> at the class level but not at the method level.
@Disabled	Used to disable a test class or test method; analogous to JUnit 4's @Ignore. Such annotations are not <i>inherited</i> .
@Timeout	Used to fail a test, test factory, test template, or lifecycle method if its execution exceeds a given duration. Such annotations are <i>inherited</i> .
@ExtendWith	Used to register extensions declaratively . Such annotations are <i>inherited</i> .
@RegisterExtension	Used to register extensions programmatically via fields. Such fields are <i>inherited</i> unless they are <i>shadowed</i> .
@TempDir	Used to supply a temporary directory via field injection or parameter injection in a lifecycle method or test method; located in the <code>org.junit.jupiter.api.io</code> package.

----- The END -----