JUnit 5 [@Tag](https://junit.org/junit5/docs/current/api/org.junit.jupiter.api/org/junit/jupiter/api/Tag.html) is used to tag or label the tests into a [test suite](https://howtodoinjava.com/junit5/junit5-test-suites-examples/). It helps organize and filter the tests based on different criteria or attributes, such as their purpose, the features they test, or their characteristics. This particularly helps in selectively running a subset of tests based on their tags, making test suite management more flexible.

We can execute a set of tests by including only those tagged tests in the test plan OR by excluding other tests from the test plan.

## **1. Using *@Tag* Annotation**

We can apply the @Tag annotation on a test class, test method, or both. We can use any string as a tag value, but it’s common to use descriptive tags like “*smoke*,” “*regression*,” “*integration*,” “*security*,” and so on.

@Tag("development")

public class ClassATest

{

@Test

@Tag("userManagement")

void testCaseA(TestInfo testInfo) {

}

}

We can apply multiple tags on a single test case as well so that you can include the test in multiple test suites.

public class ClassATest

{

@Test

@Tag("development")

@Tag("production")

void testCaseA(TestInfo testInfo) {

}

}

## **2. Creating Test Suites with *@IncludeTags* and *@ExcludeTags***

We can use [*@IncludeTags*](https://junit.org/junit5/docs/current/api/index.html?org/junit/platform/runner/IncludeTags.html) or [*@ExcludeTags*](https://junit.org/junit5/docs/current/api/index.html?org/junit/platform/runner/ExcludeTags.html) annotations in the suites to filter tests or include tests.

Take an example. In the following example, when the first suite runs then it will run all the tests tagged with “*production*” inside the package ‘*com.howtodoinjava.junit5.examples*‘. In the second example, we exclude all those tests and execute the remaining tests.

*//@IncludeTags example*

@Suite

@SelectPackages("com.howtodoinjava.junit5.examples")

@IncludeTags("production")

public class MultipleTagsExample

{

}

*//@ExcludeTags example*

@Suite

@SelectPackages("com.howtodoinjava.junit5.examples")

@ExcludeTags("production")

public class MultipleTagsExample

{

}

To add more than one tag, pass a *string array of tags* in the desired annotation.

@Suite

@SelectPackages("com.howtodoinjava.junit5.examples")

@IncludeTags({"production","development"})

public class MultipleTagsExample

{

}

# **JUnit 5 – Test Suites with Example**

Last Updated : 09 Feb, 2024

JUnit 5 encourages a modular approach to test creation with its test suites. These suites function as containers, bundling multiple test classes for easier management and collective execution within a single run. In this article, we will learn about JUnit 5 – Test Suites.

### **Test Suite**

In JUnit 5, a test suite is a way to combine multiple test classes and run them together as a single unit. The @Suite annotation on the test class is used to define a test suite. You can also include specific test classes in the suite using annotations like @SelectClasses, @SelectPackages, or @SelectClasspathRoots, that help to filter out the test classes you want to run together. Now we go through an example, to execute a test suite.

### **Example of JUnit 5 – Test Suites**

Here we take a Java program that has methods add() and subtract() to perform addition and subtraction of two numbers. We have written test classes for both methods separately as AdditionTest and SubtractionTest. Now we can run both of them collectively as a single unit, under a test suite CalculatorTestSuite which is annotated by @Suite, and then @SelectClasses which includes both our test classes.

### **Calculator.java File:**

* Java

| // Calculator.java  **public** **class** Calculator {    **public** **static** **int** add(**int** a, **int** b) {  **return** a + b;  }    **public** **static** **int** subtract(**int** a, **int** b) {  **return** a - b;  }  } |
| --- |

### **AdditionTest.java File:**

* Java

| // AdditionTest.java  **import** **static** org.junit.jupiter.api.Assertions.\*;  **import** org.junit.jupiter.api.Test;    **class** AdditionTest {    @Test  **void** testAddition() {  Calculator calculator = **new** Calculator();  assertEquals(5, calculator.add(2, 3));  }    } |
| --- |

### **SubtractionTest.java File:**

* Java

| // SubtractionTest.java  **import** **static** org.junit.jupiter.api.Assertions.\*;  **import** org.junit.jupiter.api.Test;    **class** SubtractionTest {    @Test  **void** testSubtraction()  {  Calculator calculator = **new** Calculator();  assertEquals(2, calculator.subtract(5, 3));  }  } |
| --- |

### 

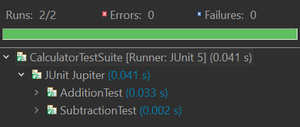
### 

### **CalculatorTestSuite.java File:**

* Java

| // CalculatorTestSuite.java  **import** org.junit.platform.suite.api.SelectClasses;  **import** org.junit.platform.suite.api.Suite;  @Suite  @SelectClasses({AdditionTest.**class**, SubtractionTest.**class**})  **public** **class** CalculatorTestSuite {  } |
| --- |

#### **Output:**

****