

assertThrows() method

This lesson demonstrates how to use assertThrows method in JUnit 5 to assert test conditions.

WE'LL COVER THE FOLLOWING ^

- assertThrows() method
- Demo

assertThrows() method

Assertions API provide static `assertThrows()` method. This method helps in asserting that execution of the supplied `Executable` throws an exception of the `expectedType` and returns the exception.

- If no exception is thrown, or if an exception of a different type is thrown, this method will fail.
- It follows the inheritance hierarchy, so the assert will pass if the expected type is `Exception` and actual is `RuntimeException`.

There are basically three useful overloaded methods for assertThrows:-

```
public static <T extends Throwable> T assertThrows(Class<T> expectedType, Executable executable) throws T { ... }  
public static <T extends Throwable> T assertThrows(Class<T> expectedType, Executable executable, long timeout) throws T { ... }  
public static <T extends Throwable> T assertThrows(Class<T> expectedType, Executable executable, long timeout, long delay) throws T { ... }
```

Demo

Let's look into the usage of the above methods:-



مشاركة



المشاهدة لاحقًا

JUnit 5 Assertions - assertThrows method



Java Unit Testing with JUnit 5

JUnit 5 Assertions – assertThrows() method

Dinesh Varyani

<https://www.hubberspot.com>

assertThrows method

```
package io.educative.junit5;

import static org.junit.jupiter.api.Assertions.assertThrows;

import java.io.IOException;

import org.junit.jupiter.api.Test;

public class AssertThrowsDemo {

    @Test
    public void testAssertThrows() {
        assertThrows(ArithmeticException.class, () -> divide(1, 0));
    }

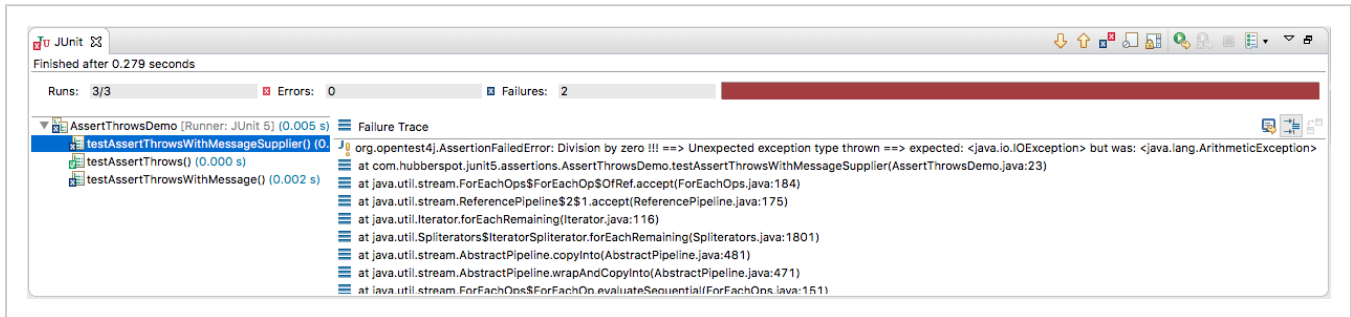
    @Test
    public void testAssertThrowsWithMessage() {
        assertThrows(IOException.class, () -> divide(1, 0), "Division by zero !!!");
    }

    @Test
    public void testAssertThrowsWithMessageSupplier() {
        assertThrows(IOException.class, () -> divide(1, 0), () -> "Division by zero !");
    }

    private int divide(int a, int b) {
        return a / b;
    }
}
```



Run AssertThrowsDemo class as JUnit Test.



In the next lesson, we will look into `assertTimeout()` and `assertTimeoutPreemptively()` assertions.