assertTimeout() method

This lesson demonstrates how to use assertTimeout method in JUnit 5 to assert timeout conditions.

We'll cover the following

assertTimeout() method

Assertions API provide static `assertTimeout()` method. It is used to test long-running tasks. If given task inside the test case takes more than the specified duration, then the test will fail.

The executable provided to the test case will be executed in the same thread as that of the calling code. Also, the execution of the executable will not be preemptively aborted if the timeout is exceeded.

There are basically three useful overloaded methods for assertTimeout:-

```
1  public static void assertTimeou
2
3  public static void assertTimeou
4
5  public static void assertTimeou
```

Demo

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

assertTimeout method

```
package io.educative.junit5;

import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.junit.jupiter.api.Assertions.assertTimeout;

import java.time.Duration;

import org.junit.jupiter.api.Test;

public class AssertTimeoutDemo {
        @Test
    void timeoutNotExceeded() {
        // The following assertion succeeds.
        assertTimeout(Duration.ofMinutes(3), () -> {
            // Perform task that takes less than 3 minutes.
        });
    }

    @Test
    void timeoutNotExceededWithResult() {
        // The following assertion succeeds, and returns the supplied object.
        String actualResult = assertTimeout(Duration.ofMinutes(3), () -> {
            return "result";
        });
        assertEquals("result", actualResult);
    }

    @Test
    void timeoutNotExceededWithMethod() {
```

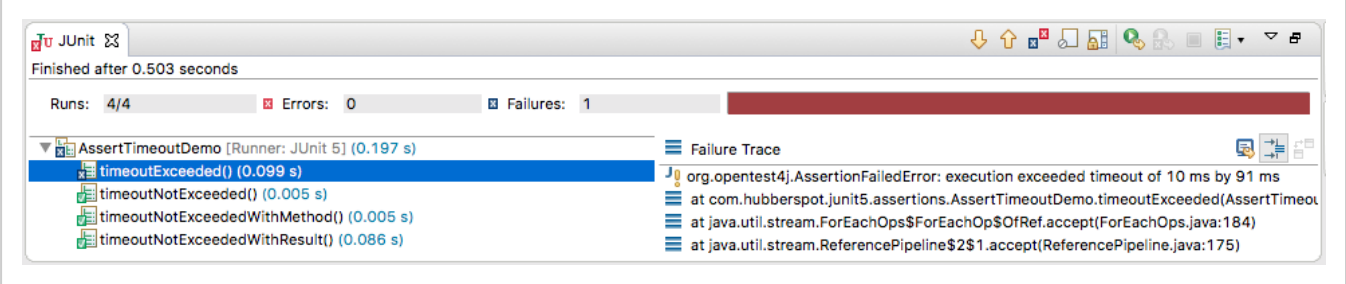
```
        // The following assertion invokes a method reference and returns an object.
        String actualGreeting = assertTimeout(Duration.ofMinutes(3), AssertTimeoutDemo::greet
        assertEquals("Hello, World!", actualGreeting);

    }

    @Test
    void timeoutExceeded() {
        // The following assertion fails with an error message similar to:
        // execution exceeded timeout of 10 ms by 91 ms
        assertTimeout(Duration.ofMillis(10), () -> {
            // Simulate task that takes more than 10 ms.
            Thread.sleep(100);
        });
    }

    private static String greeting() {
        return "Hello, World!";
    }
}
```

Run AssertTimeoutDemo class as JUnit Test.



In the next chapter, we will discuss about `assertTimeoutPreemptively` assertion.