

fail() method

This lesson demonstrates the importance of the fail method in JUnit 5 Assertions API.

We'll cover the following



- fail() method

fail() method

Assertions API provide static `fail()` method. As soon as, any `@Test` method encounters `fail()` static method, it will fail the test case. The primary usages of `fail()` method are as follows -

- It gives a piece of meaningful information to the programmer writing a test, that test case is in progress and still needs to be implemented.
- It can be used to verify that an actual exception is thrown. Usually based on some input when test case expects an exception at a certain line, providing `fail()` below that line will verify that exception was not thrown as code execution reached `fail()` method line. Thus, it explicitly fails the test case.

There are basically five useful overloaded methods to fail:-

```
1 public static void fail()
2
3 public static void fail(String
4
5 public static void fail(Supplie
6
7 public static void fail(Throwa
8
9 public static void fail(String
```



```
1 package io.educative.junit5;
2
3 import static org.junit.jupiter
4
```



```

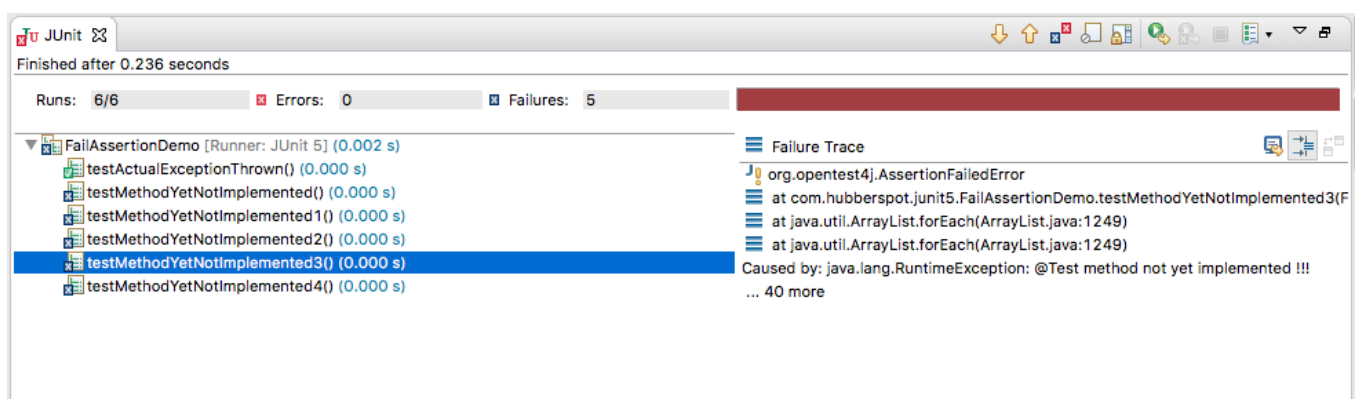
5 import org.junit.jupiter.api.*
6
7 public class FailAssertionDemo
8
9     // usage 1 - @Test not imp
10    @Test
11    public void testMethodYetNo
12        fail();
13    }
14
15    // usage 2 - @Test not imp
16    @Test
17    public void testMethodYetNo
18        fail("@Test method not
19    }
20
21    // usage 3 - @Test not imp
22    @Test
23    public void testMethodYetNo
24        fail(() -> "@Test metho
25    }
26
27    // usage 4 - @Test not imp
28    @Test
29    public void testMethodYetNo
30        fail(new RuntimeExcepti
31    }

```



You can perform code changes to above code widget, run and practice different outcomes.

Step 4 - Run `FailedAssertionDemo.java` class as Junit Test.



In the next lesson, we will look into `assertTrue()` and `assertFalse()` assertion.

