

assumeTrue() and assumeFalse() method

This lesson demonstrates how to use `assumeTrue` and `assumeFalse` methods in JUnit 5 to make conditional assumptions.

We'll cover the following ^

- `assumeTrue()`
- Demo
- `assumeFalse()`
- Demo

`assumeTrue()`

Assumptions API in JUnit 5 has a static utility method called as, `assumeTrue()`. It validates the given assumption to true.

- if the assumption is **true** then test proceeds to execution.
- if the assumption is **false** then test execution is aborted.

There are basically three useful overloaded methods for `assumeTrue`.

```
1 // boolean assumption to validate
2 public static void assumeTrue(boolean assumption) throws TestAbortedException
3 public static void assumeTrue(boolean assumption, Supplier<String> messageSupplier) thro
4 public static void assumeTrue(boolean assumption, String message) throws TestAbortedExce
5
6 // BooleanSupplier to provide boolean assumption to validate
7 public static void assumeTrue(BooleanSupplier assumptionSupplier) throws TestAbortedExce
8 public static void assumeTrue(BooleanSupplier assumptionSupplier, String message) throws
9 public static void assumeTrue(BooleanSupplier assumptionSupplier, Supplier<String> messa
```

Demo

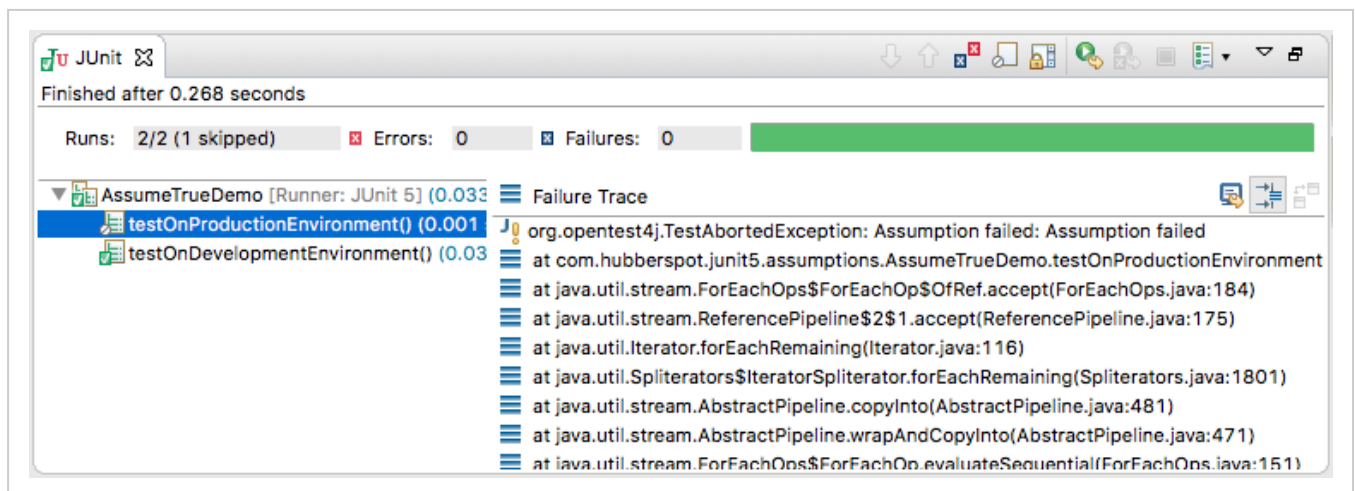
Let's look into the usage of the above methods.

```
1 package io.educative.junit5;
```

```

2
3 import static org.junit.jupiter.api.Assumptions.assumeTrue;
4
5 import org.junit.jupiter.api.Test;
6
7 public class AssumeTrueDemo {
8
9     @Test
10    void testOnDevelopmentEnvironment() {
11        System.setProperty("ENV", "DEV");
12        assumeTrue("DEV".equals(System.getProperty("ENV")));
13        //remainder of test will proceed
14    }
15
16    @Test
17    void testOnProductionEnvironment() {
18        System.setProperty("ENV", "PROD");
19        assumeTrue("DEV".equals(System.getProperty("ENV")), "Assumption failed");
20        // remainder of test will be aborted
21    }
22 }

```



assumeFalse()

Assumptions API in JUnit 5 has a static utility method called as `assumeFalse()`. It validates the given assumption to false.

- if the assumption is **false** then test proceeds to execution.
- if the assumption is **true** then test execution is aborted.

There are basically three useful overloaded methods for `assumeFalse`.

```
// boolean assumption to validate
public static void assumeFalse(boolean assumption) throws TestAbortedException
public static void assumeFalse(boolean assumption, Supplier<String> messageSupplier) throws TestAbortedException
public static void assumeFalse(boolean assumption, String message) throws TestAbortedException

// BooleanSupplier to provide boolean assumption to validate
public static void assumeFalse(BooleanSupplier assumptionSupplier) throws TestAbortedException
public static void assumeFalse(BooleanSupplier assumptionSupplier, String message) throws TestAbortedException
public static void assumeFalse(BooleanSupplier assumptionSupplier, Supplier<String> messageSupplier) throws TestAbortedException
```

Demo

Let's look into the usage of the above methods.

```
package io.educative.junit5;

import static org.junit.jupiter.api.Assumptions.assumeFalse;

import org.junit.jupiter.api.Test;

public class AssumeFalseDemo {

    @Test
    void testOnDevelopmentEnvironment() {
        System.setProperty("ENV", "DEV");
        assumeFalse("DEV".equals(System.getProperty("ENV")), "Assumption failed");
        //remainder of test will be aborted
    }

    @Test
    void testOnProductionEnvironment() {
        System.setProperty("ENV", "PROD");
        assumeFalse("DEV".equals(System.getProperty("ENV")));
        // remainder of test will proceed
    }
}
```



JUnit 5

Finished after 0.333 seconds

Runs: 2/2 (1 skipped) Errors: 0 Failures: 0

AssumeFalseDemo [Runner: JUnit 5] (0.00s)

- testOnProductionEnvironment() (0.00s)
- testOnDevelopmentEnvironment() (0.00s)

Failure Trace

```
org.opentest4j.TestAbortedException: Assumption failed: Assumption failed
    at com.hubberspot.junit5.assumptions.AssumeFalseDemo.testOnDevelopmentEnvironment(AssumeFalseDemo.java:18)
    at java.util.stream.ForEachOps$ForEachOp$OfRef.accept(ForEachOps.java:184)
    at java.util.stream.ReferencePipeline$2$1.accept(ReferencePipeline.java:175)
    at java.util.Iterator.forEachRemaining(Iterator.java:116)
    at java.util.Spliterators$IteratorSpliterator.forEachRemaining(Spliterators.java:1801)
    at java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:481)
    at java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:471)
    at java.util.stream.ForEachOps$ForEachOp.evaluateSequential(ForEachOps.java:151)
```

In the next lesson we will learn about `assumingThat()` method.