assumeTrue() and assumeFalse() method

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

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.

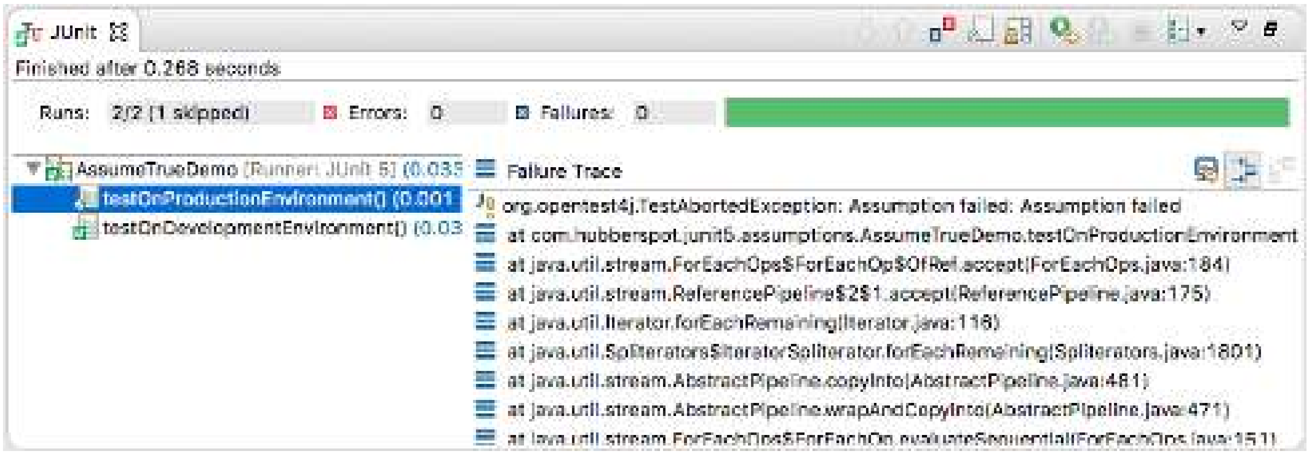
```
// boolean assumption to validate
public static void assumeTrue(boolean assumption) throws TestAbortedException
public static void assumeTrue(boolean assumption, Supplier<String> messageSupplier) throws Te
public static void assumeTrue(boolean assumption, String message) throws TestAbortedExceptio

// BooleanSupplier to provide boolean assumption to validate
public static void assumeTrue(BooleanSupplier assumptionSupplier) throws TestAbortedExceptio
public static void assumeTrue(BooleanSupplier assumptionSupplier, String message) throws Test
public static void assumeTrue(BooleanSupplier assumptionSupplier, Supplier<String> messageSup
```

Demo

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

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

```
import static org.junit.jupiter.api.Assumptions.assumeTrue;


import org.junit.jupiter.api.Test;

public class AssumeTrueDemo {

        @Test
    void testOnDevelopmentEnvironment() {
        System.setProperty("ENV", "DEV");
        assumeTrue("DEV".equals(System.getProperty("ENV")));
        //remainder of test will proceed
    }

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



```
Finished after 0.268 seconds
Runs: 2/2 (1 skipped)   Errors: 0   Failures: 0

AssumeTrueDemo [Runner: JUnit 5] (0.035
  testOnProductionEnvironment() (0.001
  testOnDevelopmentEnvironment() (0.03

Failure Trace
org.opentest4j.TestAbortedException: Assumption failed: Assumption failed
at com.hubberspot.junit5.assumptions.AssumeTrueDemo.testOnProductionEnvironment
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)
```

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.

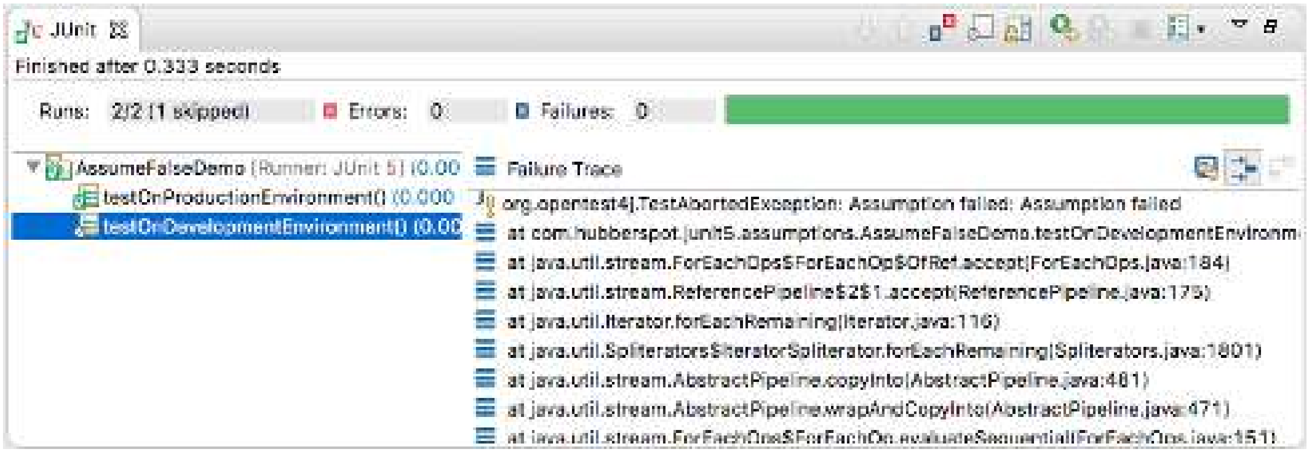
```
// boolean assumption to validate
public static void assumeFalse(boolean assumption) throws TestAbortedException
public static void assumeFalse(boolean assumption, Supplier<String> messageSupplier) throws T
public static void assumeFalse(boolean assumption, String message) throws TestAbortedExceptio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

```
// BooleanSupplier to provide boolean assumption to validate
public static void assumeFalse(BooleanSupplier assumptionSupplier) throws TestAbortedExceptic

public static void assumeFalse(BooleanSupplier assumptionSupplier, String message) throws Tes
public static void assumeFalse(BooleanSupplier assumptionSupplier, Supplier<String> messageSu
```

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
    }
}
```



```
Finished after 0.333 seconds

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

AssumeFalseDemo [Runner: JUnit 5] (0.00
    testOnProductionEnvironment() (0.000
    testOnDevelopmentEnvironment() (0.00

org.opentest4j.TestAbortedException: Assumption failed: Assumption failed
at com.hubberspot.junit5.assumptions.AssumeFalseDemo.testOnDevelopmentEnvironm
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.