JUnit 4 Test Timeout

Learn to write JUnit tests with timeout behavior. If a test does not complete execution in a given time limit then its execution will be forcefully stopped and the test will be marked failed.

**1. Why tests need to be timedout?**

We might need to test for the fail-safe behavior of the application that how the application behaves when the external systems do not respond in the given time.

That’s why it is advisable to use timeout-related tests when we are mocking the external dependencies.

This is also considered **[JUnit best practice](https://howtodoinjava.com/best-practices/unit-testing-best-practices-junit-reference-guide/" \o "Unit testing best practices : Junit Reference guide)** to be followed.

**How the timeout is invoked?**

As we know that each test is run in a new thread. If the specified timeout elapses before the test completes, its execution is interrupted via **Thread.interrupt()**.

**2. JUnit test timeout example**

To specify the **timeout** period of a certain test case, “timeout” attribute is mentioned on the annotation @Test.

Note that the timeout time is specified in **milliseconds**.

Timeout test example

@Test(timeout = 500)

**public** **void** testTimeoutOne() **throws** InterruptedException

{

TimeUnit.SECONDS.sleep(1);

}

In the above test, execution will be timed out after 500ms with the below message.

java.lang.Exception: test timed out after 500 milliseconds

**2. Timeout Rule for Global Timeouts**

Rather than specifying the *timeout* attributes for all the tests separately, we can define **JUnit Rule** for all tests in a class.

Global timeout test example

**import** java.util.concurrent.TimeUnit;

**import** org.junit.Rule;

**import** org.junit.Test;

**import** org.junit.rules.Timeout;

**public** **class** TimeoutTests {

@Rule

**public** Timeout globalTimeout = Timeout.seconds(2);

@Test

**public** **void** testTimeoutOne() **throws** InterruptedException *//PASS*

{

TimeUnit.SECONDS.sleep(1);

}

@Test

**public** **void** testTimeoutTwo() **throws** InterruptedException *//FAIL*

{

TimeUnit.SECONDS.sleep(3);

}

}

In the above example, the first test will PASS while the second test will FAIL.