

Assertions: Throws and Timeout



Assertions

Method name	Description
assertThrows	Assert that an executable throws an exception of expected type

Code to Test

DemoUtils.java

```
package com.luv2code.junitdemo;

public class DemoUtils {

    public String throwException(int a) throws Exception {
        if (a < 0) {
            throw new Exception("Value should be greater than or equal to 0");
        }
        return "Value is greater than or equal to 0";
    }
}
```

Verify this method
throws an exception for values < 0

DemoUtilsTest.java

```
package com.luv2code.junitdemo;

import org.junit.jupiter.api.*;
import static org.junit.jupiter.api.Assertions.*;

class DemoUtilsTest {

    DemoUtils demoUtils;

    ...

    @DisplayName("Throws and Does Not Throw")
    @Test
    void testThrowsAndDoesNotThrow() {
        assertThrows(Exception.class, () -> { demoUtils.throwException(-1); }, "Should throw exception");

        assertDoesNotThrow(() -> { demoUtils.throwException(5); }, "Should not throw exception");
    }
}
```

Lambda expression

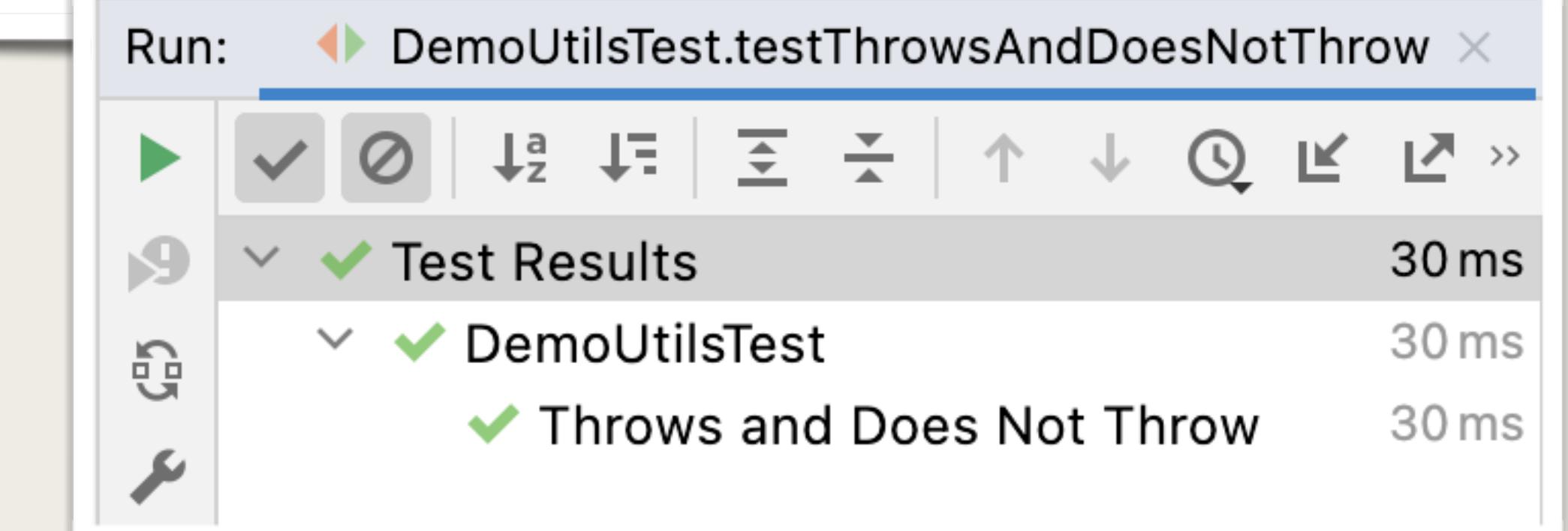
Lambda expression

DemoUtils.java

```
package com.luv2code.junitdemo;

public class DemoUtils {

    public String throwException(int a) throws Exception {
        if (a < 0) {
            throw new Exception("Value should be greater than or equal to 0");
        }
        return "Value is greater than or equal to 0";
    }
}
```



Assertions

Method name	Description
<code>assertTimeoutPreemptively</code>	Assert that an executable completes before given timeout is exceeded

**Execution is preemptively aborted
if timeout is exceeded**

Code to Test

DemoUtils.java

```
package com.luv2code.junitdemo;

public class DemoUtils {

    public void checkTimeout() throws InterruptedException {
        System.out.println("I am going to sleep");
        Thread.sleep(2000);
        System.out.println("Sleeping over");
    }
}
```

Make sure method doesn't oversleep. :-)

DemoUtilsTest.java

```
package com.luv2code.junitdemo;  
  
import org.junit.jupiter.api.*;  
import static org.junit.jupiter.api.Assertions.*;  
import java.time.Duration;  
  
class DemoUtilsTest {  
  
    DemoUtils demoUtils;  
  
    ...  
  
    @DisplayName("Timeout")  
    @Test  
    void testTimeout() {  
  
        assertTimeoutPreemptively(Duration.ofSeconds(3), () -> { demoUtils.checkTimeout(); },  
            "Method should execute in 3 seconds");  
    }  
}
```

Timeout duration

Lambda expression

DemoUtils.java

```
package com.luv2code.junitdemo;  
  
public class DemoUtils {  
  
    public void checkTimeout() throws InterruptedException {  
        System.out.println("I am going to sleep");  
        Thread.sleep(2000);  
        System.out.println("Sleeping over");  
    }  
}
```

