

# Code Coverage and Test Reports with Maven



# Code Coverage Reports

- If you are not using IntelliJ, don't worry
  - In this video, I'll show you how to do the same using Maven commands
  - No IDE is required ... run Maven from command-line
  - Useful when running as part of DevOps build process
  - Continuous Integration / Continuous Deployment (CI/CD) environments

# System Requirements

- To complete these steps, you must have Maven installed (not with IDE)
- If you don't have Maven installed, follow steps at: <https://maven.apache.org>
- Once installed, verify your Maven installation
- Open a terminal window and type: **mvn -version**



```
Apache Maven 3.8.4 (9b656c72d54e5bacbed989b64718c159fe39b537)
```

```
Maven home: ...
```

```
...
```

# Development Process

*Step-By-Step*

1. Configure Maven to find unit tests
2. Run unit tests
3. Generate unit test reports
4. Generate code coverage reports

# Step 1: Configure Maven to find unit tests

- By default, Maven will NOT find JUnit 5 test classes
- At command-line, type: **mvn clean test**



```
...
-----
T E S T S
-----
Results :
Tests run: 0, Failures: 0, Errors: 0, Skipped: 0
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.450 s
[INFO] Finished at: 2022-1-2T10:20:17-05:00
[INFO] -----
```

No tests were found??!!!

# Step 1: Configure Maven to find unit tests

- Resolve this by using Maven **Surefire** Plugin version 2.22.0 or higher

pom.xml

```
...
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>3.0.0-M5</version>
    </plugin>
  </plugins>
</build>
...
```

# Step 2: Run unit tests

- At command-line, type: **mvn clean test**

```
...
[INFO] --- maven-surefire-plugin:3.0.0-M5:test (default-test) @ junitdemo ---
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.luv2code.junitdemo.DemoUtilsTest
I am going to sleep
Sleeping over
[INFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.077 s - in com.luv2code.junitdemo.DemoUtilsTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 4.673 s
[INFO] Finished at: 2022-1-2T02:32:07-05:00
[INFO] -----
```

# Step 3: Generate unit test reports

- Maven **SureFire-Report** plugin can generate HTML unit test report

pom.xml

```
...
<build>
  <plugins>

    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-report-plugin</artifactId>
      <version>3.0.0-M5</version>

      <!-- TO DO ... -->

    </plugin>

  </plugins>
</build>
...
```

# Step 3: Generate unit test reports

pom.xml

```
...
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-report-plugin</artifactId>
      <version>3.0.0-M5</version>
      <executions>
        <execution>
          <phase>test</phase>
          <goals>
            <goal>report</goal>
          </goals> _____
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

During Maven's test phase,

execute the plugin goal

**maven-surefire-report-plugin:report**

# Step 3: Generate unit test reports

- At command-line, type:

```
mvn clean test
```

Run tests and  
executes Surefire Report Plugin  
to generate HTML reports

```
mvn site -DgenerateReports=false
```

Add website resources  
images, css etc...

Don't overwrite existing HTML reports

# View unit test reports

File: target/site/surefire-report.html

## Surefire Report

### Summary

[Summary] [Package List] [Test Cases]

Tests	Errors	Failures	Skipped	Success Rate	Time
9	0	0	0	100%	2.069

Note: failures are anticipated and checked for with assertions while errors are unanticipated.

### Package List

[Summary] [Package List] [Test Cases]

Package	Tests	Errors	Failures	Skipped	Success Rate	Time
com.luv2code.junitdemo	9	0	0	0	100%	2.069

Note: package statistics are not computed recursively, they only sum up all of its testsuites numbers.

### com.luv2code.junitdemo

	Class	Tests	Errors	Failures	Skipped	Success Rate	Time
⚠	DemoUtilsTest	9	0	0	0	100%	2.069

## Test Cases

[Summary] [Package List] [Test Cases]

### DemoUtilsTest

⚠	testSameAndNotSame	0.021
⚠	testArrayEquals	0.001
⚠	testTrueFalse	0.002
⚠	testNullAndNotNull	0.002
⚠	testEqualsAndNotEquals	0.001
⚠	testTimeout	2.01
⚠	testLinesMatch	0.004
⚠	testIterableEquals	0.002
⚠	testThrowsAndDoesNotThrow	0.003

# Handling tests failure

- By default, Maven **Surefire** plugin will NOT generate reports if tests fail
- To generate reports if tests pass or fail

pom.xml

```
...
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>3.0.0-M5</version>

      <configuration>
        <testFailureIgnore>true</testFailureIgnore>
      </configuration>
    </plugin>
  </plugins>
</build>
...
```

generate reports  
if tests pass or fail

# Show Failing Tests

File: target/site/surefire-report.html

## Surefire Report

### Summary

[Summary] [Package List] [Test Cases]

Tests	Errors	Failures	Skipped	Success Rate	Time
9	0	1	0	88.889%	2.073

Note: failures are anticipated and checked for with assertions while errors are unanticipated.

### Package List

[Summary] [Package List] [Test Cases]

Package	Tests	Errors	Failures	Skipped	Success Rate	Time
com.luv2code.junitdemo	9	0	1	0	88.889%	2.073

Note: package statistics are not computed recursively, they only sum up all of its testsuites numbers.

### com.luv2code.junitdemo

	Class	Tests	Errors	Failures	Skipped	Success Rate	Time
⚠	DemoUtilsTest	9	0	1	0	88.889%	2.073

## Test Cases

[Summary] [Package List] [Test Cases]

### DemoUtilsTest

⚠	testSameAndNotSame	0.021
⚠	testArrayEquals	0.001
✖	testTrueFalse + [ Detail ]	0.003
⚠	testNullAndNotNull	0.002
⚠	testEqualsAndHashCode	0.001
⚠	testTimeout	2.01
⚠	testLinesMatch	0.005
⚠	testIterableEquals	0.002
⚠	testThrowsAndDoesNotThrow	0.002

## Failure Details

[Summary] [Package List] [Test Cases]

✖	testTrueFalse	
	org.opentest4j.AssertionFailedError	
	com.luv2code.junitdemo.DemoUtilsTest:60	

# Showing @DisplayName in reports

- By default, Maven **Surefire** plugin will NOT show @DisplayName in reports
- To show @DisplayName in reports

pom.xml

```
...
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>3.0.0-M5</version>

      <configuration>
        <testFailureIgnore>true</testFailureIgnore>
        <statelessTestsetReporter implementation="org.apache.maven.plugin.surefire.extensions.junit5.JUnit5Xml130StatelessReporter">
          <usePhrasedTestCaseMethodName>true</usePhrasedTestCaseMethodName>
        </statelessTestsetReporter>
      </configuration>
    </plugin>
  </plugins>
</build>
...
```

**Show @DisplayName**

# Showing @DisplayName

File: target/site/surefire-report.html

**Surefire Report**

**Summary**

[Summary] [Package List] [Test Cases]

Tests	Errors	Failures	Skipped	Success Rate	Time
9	0	0	0	100%	2.069

Note: failures are anticipated and checked for with assertions while errors are unanticipated.

**Package List**

[Summary] [Package List] [Test Cases]

Package	Tests	Errors	Failures	Skipped	Success Rate	Time
com.luv2code.junitdemo	9	0	0	0	100%	2.069

Note: package statistics are not computed recursively, they only sum up all of its testsuites numbers.

**com.luv2code.junitdemo**

	Class	Tests	Errors	Failures	Skipped	Success Rate	Time
⚠	DemoUtilsTest	9	0	0	0	100%	2.069

**Showing @DisplayName**

**Test Cases**

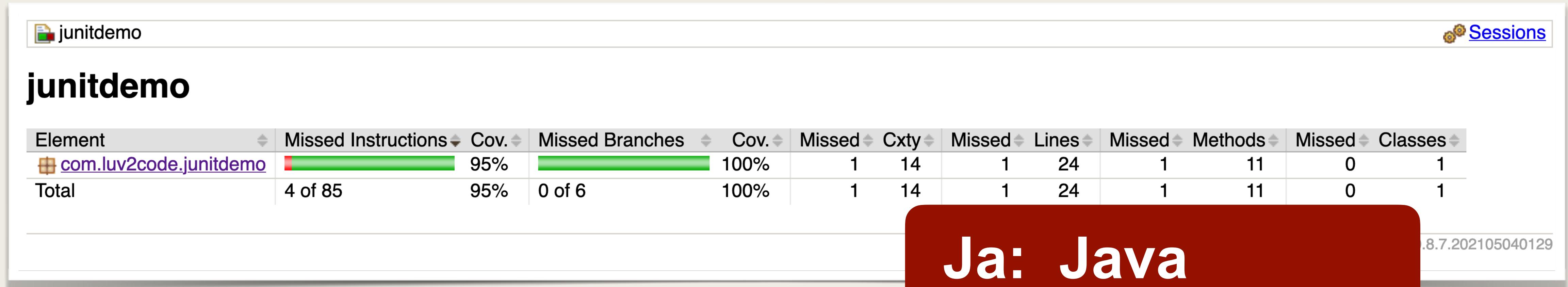
[Summary] [Package List] [Test Cases]

**DemoUtilsTest**

⚠	Same and Not Same	0.032
⚠	Array Equals	0.006
⚠	True and False	0.003
⚠	Null and Not Null	0.002
⚠	Equals and Not Equals	0.004
⚠	Timeout	2.009
⚠	Lines match	0.006
⚠	Iterable equals	0.003
⚠	Throws and Does Not Throw	0.004

# Step 4: Generate code coverage reports

- JaCoCo is a free code coverage library
- JaCoCo provides a Maven plugin to generate code coverage reports



[www.jacoco.org](http://www.jacoco.org)

Ja: Java  
Co: Code  
Co: Coverage

# Step 4: Generate code coverage reports

pom.xml

```
...
<build>
  <plugin>
    <groupId>org.jacoco</groupId>
    <artifactId>jacoco-maven-plugin</artifactId>
    <version>0.8.7</version>
    <executions>
      <execution>
        <id>jacoco-prepare</id>
        <goals>
          <goal>prepare-agent</goal>
        </goals>
      </execution>
      <execution>
        <id>jacoco-report</id>
        <phase>test</phase>
        <goals>
          <goal>report</goal>
        </goals>
      </execution>
    </executions>
  </plugin>
</build>
...
```

Prepare JaCoCo agent

This goal is bound by default to  
Maven's initialize phase

During Maven's test phase,

execute the plugin goal

**jacoco-maven-plugin:report**

# Step 4: Generate code coverage reports

- At command-line, type: **mvn clean test**

```
...
[INFO] -----< com.luv2code:junitdemo >-----
[INFO] Building junitdemo 1.0
[INFO] ----- [ jar ] -----
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ junitdemo ---
[INFO] ...
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.7:prepare-agent (jacoco-prepare) @ junitdemo ---
[INFO] ...
[INFO]
[INFO] --- maven-surefire-plugin:3.0.0-M5:test (default-test) @ junitdemo ---
[INFO] ...
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.7:report (jacoco-report) @ junitdemo ---
[INFO] ...
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 5.548 s
[INFO] Finished at: 2022-1-2T04:19:16-05:00
[INFO]
...
...
```

Code generation is  
in the test phase.

Automatically generated  
tests are executed

# View code coverage reports

File: target/site/jacoco/index.html

junitdemo Sessions

## junitdemo

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxtx	Missed	Lines	Missed	Methods	Missed	Classes
<a href="#">com.luv2code.junitdemo</a>	95%	100%	1	14	1	24	1	11	0	1	0	1
Total	4 of 85	95%	0 of 6	100%	1	14	1	24	1	11	0	1

junitdemo > com.luv2code.junitdemo > DemoUtils Sessions

## DemoUtils

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxtx	Missed	Lines	Missed	Methods	Missed	Methods
<a href="#">multiply(int, int)</a>	0%	n/a	1	1	1	1	1	1	1	1	1	1
<a href="#">DemoUtils()</a>	100%	n/a	0	1	0	5	0	1	0	1	0	1
<a href="#">isGreater(int, int)</a>	100%	100%	0	2	0	3	0	3	0	1	0	1
<a href="#">throwException(int)</a>	100%	100%	0	2	0	3	0	3	0	1	0	1
<a href="#">checkTimeout()</a>	100%	n/a	0	1	0	4	0	4	0	1	0	1
<a href="#">checkNull(Object)</a>	100%	100%	0	2	0	3	0	3	0	1	0	1
<a href="#">add(int, int)</a>	100%	n/a	0	1	0	1	0	1	0	1	0	1
<a href="#">getAcademyInList()</a>	100%	n/a	0	1	0	1	0	1	0	1	0	1
<a href="#">getAcademy()</a>	100%	n/a	0	1	0	1	0	1	0	1	0	1
<a href="#">getAcademyDuplicate()</a>	100%	n/a	0	1	0	1	0	1	0	1	0	1
<a href="#">getFirstThreeLettersOfAlphabet()</a>	100%	n/a	0	1	0	1	0	1	0	1	0	1
Total	4 of 85	95%	0 of 6	100%	1	14	1	24	1	11	0	11

Created with JaCoCo 0.8.7.202105040129

# Maven Plugin Documentation

<https://maven.apache.org/plugins>

<https://www.jacoco.org/jacoco/trunk/doc/>