



[ANDROID](#) |
 [CORE JAVA](#) |
 [DESKTOP JAVA](#) |
 [ENTERPRISE JAVA](#) |
 [JAVA BASICS](#) |
 [JVM LANGUAGES](#) |
 [SOFTWARE DEVELOPMENT](#) |
 [DEVOPS](#)

△ Home » Core Java » junit » JUnit Report Generation Example

ABOUT VINOD KUMAR KASHYAP



Vinod is Sun Certified and love to work in Java and related technologies. Having more than 10 years of experience, he had developed software's including technologies like Java, Hibernate, Struts, Spring, HTML 5, jQuery, CSS, Web Services, MongoDB, AngularJS. He is also a JUG Leader of Chandigarh Java User Group.



JUnit Report Generation Example

□ Posted by: Vinod Kumar Kashyap □ in junit □ March 22nd, 2017

In this example we shall show users how we can generate reports using the Maven and JUnit. JUnit Report Generation example demonstrates the basic usage of the reporting functionality of JUnit tests.

As you already know, JUnit is the basic unit test framework for the Java programmers. This example focuses more on generating the report. Let's start by analyzing the ways through which we can generate the HTML reports of our test cases.

1. Introduction

JUnit helps us in validation our methods for functionality. But in some cases we have to see the report also for the test cases. In the process of developing reports, Maven plays an important role as it will make a text, XML and also HTML reports. All JUnit

test cases with the details are printed in the reports. We will see in this example how this can be achieved.

However, reports can be generated in many different ways like with Ant, TestNG and other independent libraries. But we will focus on very simple case i.e. with the help of Maven.

We will be using the **surefire** plugin of maven to generate the reports for our example.

Want to be a JUnit Master ?

Subscribe to our newsletter and download the JUnit Programming Cookbook [right now!](#)

In order to help you master unit testing with JUnit, we have compiled a kick-ass guide with all the major JUnit features and use cases! Besides studying them online you may download the eBook in PDF format!

Email address:

[Sign up](#)

2. Technologies Used

We will be using following technologies to work n this example

CARRER OPPORTUNITIES



Get the latest jobs to your ir

[Send me jobs!](#)

[Job Search](#)



- **Maven** – dependency management tool and also to generate reports for our test cases.
- **JUnit 4.12** – testing framework

3. Project Setup

Tip
You may skip project creation and jump directly to the **beginning of the example** below.
Open Eclipse. Select

File -> New -> Maven Project

. Following screen will appear. Fill in the values displayed below and then click on Next.

Figure 1: JUnit Report Generation Example Setup 1

Fill in all the details as shown and click on Finish button.

NEWSLETTER

182,134 insiders are already enjoying weekly updates and complimentary whitepapers!
Join them now to gain **EXCLUSIVE ACCESS** to the latest news in the Java world as well as insights about Android, Scala, Groovy and other related technologies.

Email address:

Your email address

☒ Receive Java & Developer job alerts in your Area from our partners over at ZipRecruiter

Sign up

JOIN US



With **1,240,600** unique visitors and **500** authors placed among the top related sites at Google, we are constantly being looked out for and encourage you to join. So If you have unique and interesting content then you can check out our **JCG** partners program. You can be a **guest writer** for Java Code Geek and showcase your writing skills!



Figure 2: JUnit Report Generation Example Setup 2

Clicking on Finish button will create a blank Maven project that will be a starting point of our example.

4. JUnit Report Generation Example

First of all we need to put the dependencies for our project. Simply put the following line in the

pom.xml

file.

[pom.xml](#)

```
01 <dependencies>
02     <dependency>
03         <groupId>junit</groupId>
04         <artifactId>junit</artifactId>
05         <version>4.12</version>
06     </dependency>
07 </dependencies>
08
09 <properties>
10     <maven.compiler.source>1.8</maven.compiler.source>
11     <maven.compiler.target>1.8</maven.compiler.target>
12 </properties>
13
14 <reporting>
15     <plugins>
16         <plugin>
17             <groupId>org.apache.maven.plugins</groupId>
18             <artifactId>maven-surefire-report-plugin</artifactId>
19             <version>2.19.1</version>
20         </plugin>
21     </plugins>
22 </reporting>
```

Lines 1-7 will download JUnit jar file.

Lines 9-12 will ask Maven to use Java 1.8 for this example.

Lines 14-22 are used to fetch the surefire plugin that helps us to generate the report for our test cases.

This will ready our project. Let's start creating a unit test case.

generated.

JUnitReportGenerationTest.java

```
01 package junitreportgeneration;
02
03 import static org.hamcrest.CoreMatchers.instanceOf;
04 import static org.hamcrest.CoreMatchers.is;
05 import static org.junit.Assert.assertThat;
06 import static org.junit.Assert.assertTrue;
07
08 import java.util.ArrayList;
09 import java.util.List;
10
11 import org.junit.Test;
12
13 public class JUnitReportGenerationTest {
14
15     private String developer = "Vinod";
16
17     @Test
18     public void instanceOfTest() {
19         assertThat(new ArrayList(), instanceOf(List.class));
20     }
21
22     @Test
23     public void assertTrueTest() {
24         assertTrue(true);
25     }
26
27     @Test
28     public void equalToTest() {
29         assertThat(developer, is("Vinod"));
30     }
31
32     @Test
33     public void failTest() {
34         assertThat(developer, is("Any"));
35     }
36 }
```

This is a simple test class.

5. Generate Reports

To generate a report you need to simple run the Maven command:

```
mvn clean install test surefire-report:report
```

To run from eclipse you need to follow some steps.

1. Right click on project
2. Run As -> Run Configurations...
3. You will be prompted with the screen