



[△ Home](#) » [Core Java](#) » [junit](#) » JUnit Integration Test Example

ABOUT ALVIN REYES



Alvin has an Information Technology Degree from Mapua Institute of Technology. During his studies, he was already heavily involved in a number of small to large projects where he primarily contributes by doing programming, analysis design. After graduating, he continued to do side projects on Mobile, Desktop and Web Applications.



JUnit Integration Test Example

□ Posted by: Alvin Reyes □ in junit □ August 28th, 2015

1. Introduction

Integration tests are test cases that test highly coupled external services. A great example of this is services in a SOA environment. In this scheme we will create services (or micro services) that are usually deployed on a different container and only exposed specific implementation of it's own, consumed by a more sophisticated system.

By rule of thumb, we always need to separate this kind of tests from the internal service of the application. This separation, in JUnit Test case perspective, is called "categorising", a pretty straight forward definition, we categorise a specific test case by creating a mark on them using Java Interface class. In this example, I'll be giving a simple example of how we can write and categorise test cases by using the

@Category

annotation.

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:

Your email address

Sign up

2. Source

The following codes are the source codes made for this example. Link to this project can be found at the end of this article.

2.1 Java Test Interface

file:///Users/chaklader/Documents/Education/Software%20Testing/JUnit/Integration_Test.htm

NEWSLETTER

180,103 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

Sign up

JOIN US



With **1,240,650** unique visitors **500** authors placed among related sites at Constantly being looked for par encourage you So if you have unique and interesting content then you

We first need to mark our test cases. This is for us to categorise the test case that it's an integration test.

[IntegrationTest.java](#)

```
1 package com.aresyl.jgc.junit.integ;
2 public interface IntegrationTest {}
```

2.2 Test Case

After that, we will use the interface class to mark our test case as an Integration test.

[JUnitAssertExampleTests.java](#)

```
01 package com.aresyl.jgc.junit.integ;
02
03 import org.junit.Before;
04 import org.junit.Test;
05 import org.junit.experimental.categories.Category;
06
07 import com.aresyl.jgc.junit.integ.JUnitAssertEqualsServiceExample;
08 import com.aresyl.jgc.junit.integ.ServiceObject;
09
10 import static org.junit.Assert.assertEquals;
11 @Category(IntegrationTest.class)
12 public class IntegrationTestSampleTest {
13
14     private JUnitAssertEqualsServiceExample junitAssertEqualsServiceSample;
15     private ServiceObject serviceObject;
16     @Before
17     public void setData() {
18         serviceObject = new ServiceObject();
19         junitAssertEqualsServiceSample = new JUnitAssertEqualsServiceExample();
20         junitAssertEqualsServiceSample.initiateMetaData(serviceObject);
21     }
22
23     @Test
24     public void testAssertEqualsFalse() {
25         // processed the item
26         ServiceObject newServiceObject = new ServiceObject();
27         junitAssertEqualsServiceSample.initiateMetaData(newServiceObject);
28         junitAssertEqualsServiceSample.processObject(serviceObject);
29         assertEquals(serviceObject, newServiceObject);
30     }
31
32     @Test
33     public void testAssertEquals() {
34         junitAssertEqualsServiceSample.processObject(serviceObject);
35         assertEquals(serviceObject, this.serviceObject);
36     }
37
38     @Test
39     public void testAssertEqualsWithMessage() {
40         junitAssertEqualsServiceSample.processObject(serviceObject);
41         assertEquals(
42             "Same Object",
43             serviceObject, serviceObject);
44     }
45
46     @Test
47     public void testAssertEqualsFalseWithMessage() {
48         ServiceObject newServiceObject = new ServiceObject();
49         junitAssertEqualsServiceSample.postProcessing(serviceObject);
50         assertEquals(
51             "Not the Same Object",
52             newServiceObject, serviceObject);
53     }
54 }
```

check out our **JUG** partners program. 1
be a **guest writer** for Java Code Geek
your writing skills!

2.3 Maven Config

To ensure that this class will only be executed upon running the integration-test, we need to add the following configuration on our pom.xml

```
01 <build>
02     <plugins>
03         <plugin>
04             <artifactId>maven-failsafe-plugin</artifactId>
05             <version>2.12</version>
06             <dependencies>
07                 <dependency>
08                     <groupId>org.apache.maven.surefire</groupId>
09                     <artifactId>surefire-junit47</artifactId>
10                     <version>2.12</version>
11                 </dependency>
12             </dependencies>
13             <configuration>
14                 <groups>com.aresyl.jgc.junit.integ.IntegrationTest</groups>
15             </configuration>
16             <executions>
17                 <execution>
18                     <goals>
19                         <goal>integration-test</goal>
20                     </goals>
21                     <configuration>
22                         <includes>
23                             <include>/**/*.class</include>
24                         </includes>
25                     </configuration>
26                 </execution>
27             </executions>
```

```
28         </plugin>
29     </plugins>
30 </build>
```

We added the configuration that all test cases, which are categorised under Integrate test, will be part of the integration-test goal.

2.4 Maven command

Run the following:

```
mvn integration-test
```

3. Output

Here's the output of our log file:

```
01 -----
02  T E S T S
03 -----
04 Running com.areyes1.jgc.junit.integ.IntegrationTestSampleTest
05 Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.073 sec
06
07 Results:
08
09 Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
10
11 [INFO] -----
12 [INFO] BUILD SUCCESS
13 [INFO] -----
14 [INFO] Total time: 1.843 s
15 [INFO] Finished at: 2015-08-27T18:54:16+08:00
16 [INFO] Final Memory: 9M/107M
17 [INFO] -----
```

4. Download the Eclipse project

This was an example of JUnit Integration Tests.

Download

You can download the full source code of this example here: [junit-integration-tests](#)

Tagged with: [INTEGRATION](#) [INTEGRATION-TEST](#) [JUNIT](#)

Do you want to know how to develop your skillset to become a **Java Rockstar?**

Subscribe to our newsletter to start Rocking [right now!](#)

To get you started we give you our best selling eBooks for **FREE!**

1. JPA Mini Book
2. JVM Troubleshooting Guide
3. JUnit Tutorial for Unit Testing
4. Java Annotations Tutorial
5. Java Interview Questions
6. Spring Interview Questions
7. Android UI Design

and many more

Email address:

[Sign up](#)