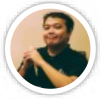
ABOUT ALVIN REYES



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.

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

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

IntegrationTest.java

```
package com.areyes1.jgc.junit.integ;
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

```
package com.areyes1.jgc.junit.integ;

import org.junit.Before;
import org.junit.Test;
import org.junit.experimental.categories.Category;

import com.areyes1.jgc.junit.integ.JUnitAssertEqualsServiceExample;
import com.areyes1.jgc.junit.integ.ServiceObject;

import static org.junit.Assert.assertEquals;
@Category(IntegrationTest.class)
public class IntegrationTestSampleTest {

    private JUnitAssertEqualsServiceExample junitAssertEqualsServiceSample;
    private ServiceObject serviceObject;
    @Before
    public void setData() {
        serviceObject = new ServiceObject();
        junitAssertEqualsServiceSample = new JUnitAssertEqualsServiceExample();
        junitAssertEqualsServiceSample.initiateMetaData(serviceObject);
    }

    @Test
    public void testAssertEqualsFalse() {
        //  processed the item
        ServiceObject newServiceObject = new ServiceObject();
        junitAssertEqualsServiceSample.initiateMetaData(newServiceObject);
        junitAssertEqualsServiceSample.processObject(serviceObject);
        assertEquals(serviceObject,newServiceObject);
    }

    @Test
    public void testAssertEquals() {
        junitAssertEqualsServiceSample.processObject(serviceObject);
        assertEquals(serviceObject,this.serviceObject);
    }

    @Test
    public void testAssertEqualsWithMessage() {
        junitAssertEqualsServiceSample.processObject(serviceObject);
        assertEquals(
                "Same Object",
                serviceObject,serviceObject);
    }
    @Test
    public void testAssertEqualsFalseWithMessage() {
        ServiceObject newServiceObject = new ServiceObject();
        junitAssertEqualsServiceSample.postProcessing(serviceObject);
        assertEquals(
                "Not the Same Object",
                newServiceObject,serviceObject);
    }
}
```

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

```
<build>
        <plugins>
            <plugin>
                <artifactId>maven-failsafe-plugin</artifactId>
                <version>2.12</version>
                <dependencies>
                    <dependency>
                        <groupId>org.apache.maven.surefire</groupId>
                        <artifactId>surefire-junit47</artifactId>
                        <version>2.12</version>
                    </dependency>
                </dependencies>
                <configuration>
                    <groups>com.areyes1.jgc.junit.integ.IntegrationTest</groups>
                </configuration>
                <executions>
                    <execution>
                        <goals>
                            <goal>integration-test</goal>
                        </goals>
                        <configuration>
                            <includes>
                                <include>**/*.class</include>
                            </includes>
                        </configuration>
                    </execution>
                </executions>
```

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

```
            </plugin>
        </plugins>
    </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:

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running com.areyes1.jgc.junit.integ.IntegrationTestSampleTest
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.073 sec

Results:

Tests run: 4, Failures: 0, Errors: 0, Skipped: 0

[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 1.843 s
[INFO] Finished at: 2015-08-27T18:54:16+08:00
[INFO] Final Memory: 9M/107M
[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