Summary

This guide explains how to group test cases that was previously written for test automation purposes.

Description

TestSuite is a JUnit 3.8 class that was provided for grouping and batch-running test cases.
JUnit 4 does not natively support TestSuite anymore, so let us merely consider what it stood for, in terms of how it can group multiple test cases together depending on their characteristics.
Accordingly, you can still write TestSuite class, but it is seldom used in the industry, and instead build tools are more of a common choice.

✔Usually, TestSuite is written for grouping existing test cases, and if the development phase moves past the unit test and moves into integration testing, it is often used for setting dependencies between test cases. This means that the unit tests are not grouped with TestSuites for actual testing in this scenario. For example, a tool called TestNG [http://testng.org/] provides skills such as by grouping test cases for integration testing.

Environmental settings

- Install the JUnit Eclipse Plugin
- Include a library for creating TestSuite (pom.xml): refer to How to modify pom.xml dependency for unit tests

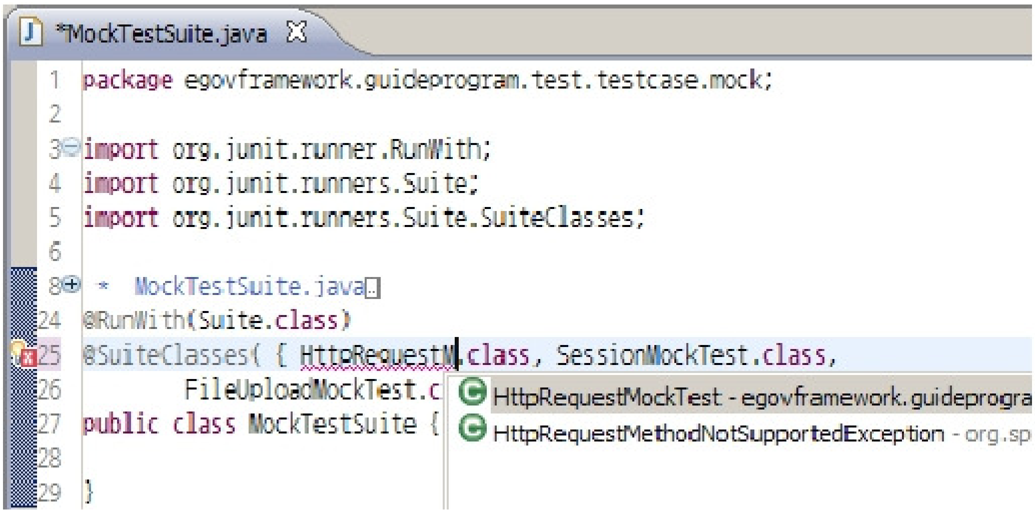
Manual

There are two ways to create TestSuites.

Creating TestSuite Class

Following is how to create TestSuite using JUnit4.

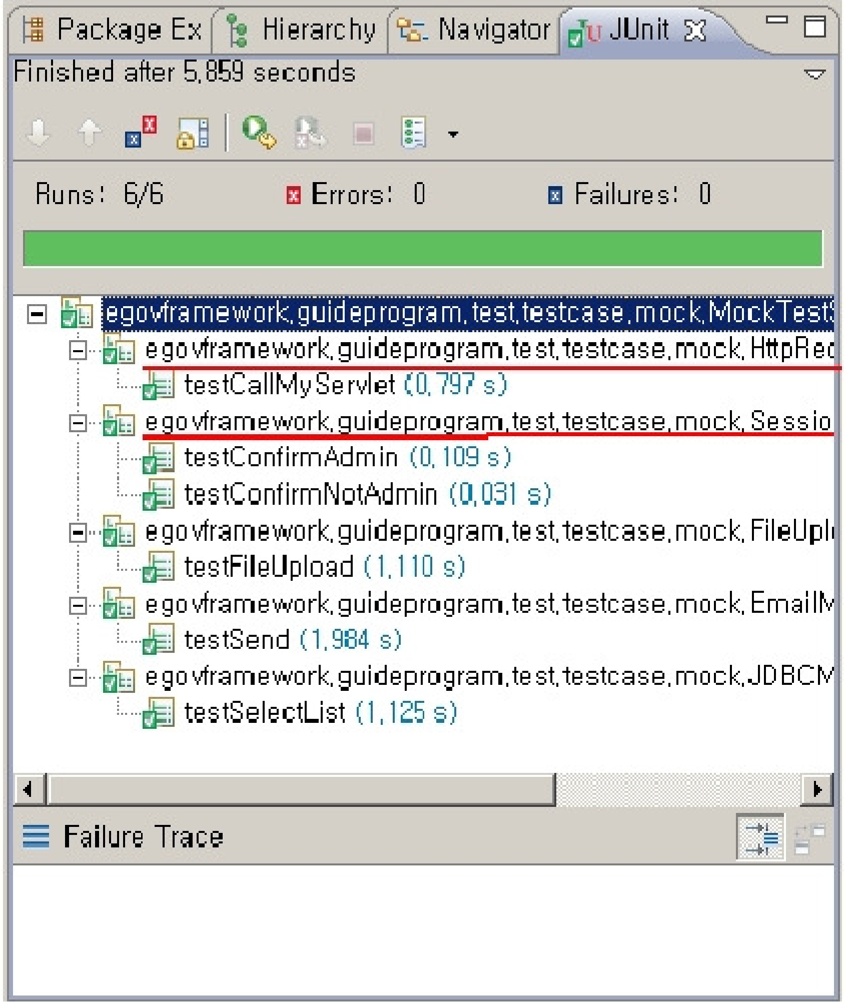
1. Create any Java class in Eclipse.[1)]
2. Enter the list of previously created test case classes as below.



```
package egovframework.guideprogram.test.testcase.mock;

import org.junit.runner.RunWith;
import org.junit.runners.Suite;
import org.junit.runners.Suite.SuiteClasses;

 * MockTestSuite.java
@RunWith(Suite.class)
@SuiteClasses( { HttpRequestM.class, SessionMockTest.class,
        FileUploadMockTest.c
public class MockTestSuite {

}
```

HttpRequestMockTest - egovframework.guideprogra
HttpRequestMethodNotSupportedException - org.sp

3. Once you are done implementing, run exactly the same way as Running Test Cases.
   If you run TestSuite as below, multiple test cases can be ran in batch.



Finished after 5.859 seconds

Runs: 6/6 Errors: 0 Failures: 0

```
egovframework.guideprogram.test.testcase.mock.MockTest
  egovframework.guideprogram.test.testcase.mock.HttpRec
    testCallMyServlet (0.797 s)
  egovframework.guideprogram.test.testcase.mock.Sessio
    testConfirmAdmin (0.109 s)
    testConfirmNotAdmin (0.031 s)
  egovframework.guideprogram.test.testcase.mock.FileUpl
    testFileUpload (1.110 s)
  egovframework.guideprogram.test.testcase.mock.EmailM
    testSend (1.984 s)
  egovframework.guideprogram.test.testcase.mock.JDBCM
    testSelectList (1.125 s)
```

✔ TestSuites are usually employed for grouping existing test cases, but also for running integration tests beyond unit tests, and you need to set dependencies between test cases. If you are only running up to unit tests then TestSuites are usually not necessary.

Writing TestSuites with build tools

You can write TestSuites using the build tools to set test cases to run.

- Maven - pom.xml :refer to Writing Test Suit in Maven
- Ant - build.xml : you can group test cases using JUnit tasks and <batchtest/> together in Ant.

Sample

TestSuite class example

```
@RunWith(Suite.class)
@SuiteClasses( { HttpRequestMockTest.class, SessionMockTest.class,
        FileUploadMockTest.class, EmailMockTest.class, JDBCMockTest.class })
public class MockTestSuite {

}
```

Maven Test Suite example

```
<build>
    <plugins>
        <!-- test -->
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-surefire-plugin</artifactId>
            <configuration>
            . . .
                <excludes>
                    <exclude>**/Abstract*.java</exclude>
                </excludes>
                <includes>
                    <include>**/*Test.java</include>
                </includes>
            </configuration>
        </plugin>
    </plugins>
</build>
```

Ant Test Suite example

```
<junit ... >
    . . . omitted . . .
    <batchtest fork="yes" todir="${testreports.dir}">
        <fileset dir="${testbuild.dir}">
             <include name="**/*Test.class" />
             <exclude name="**/Abstract*Test.class" />
             <exclude name="**/*SuiteTest.class" />
        </fileset>
    </batchtest>
</junit>
```

References

- JUnit Home : http://junit.org/ [http://junit.org/]
- JUnit FAQ : http://junit.sourceforge.net/doc/faq/faq.htm [http://junit.sourceforge.net/doc/faq/faq.htm]
- Maven Surefire Plugin : http://maven.apache.org/plugins/maven-surefire-plugin/ [http://maven.apache.org/plugins/maven-surefire-plugin/]

1) If you choose New > Others… > JUnit Test Suite, TestSuite with JUnit3 version will be created.