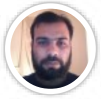
⌂ Home » Core Java » PowerMockito » PowerMockito Spy Example

ABOUT MOHAMMAD MERAJ ZIA



I did my Engineering in Information Technology from IET, Lucknow, India. Currently doing MSc in Information Technology from Derby University. I have worked in Java/J2EE domain for the last 10 years. Have good understanding of Payment and Finance domains.

PowerMockito Spy Example

□ Posted by: Mohammad Meraj Zia □ in PowerMockito □ May 4th, 2016

A unit test should test a class in isolation. Side effects from other classes or the system should be eliminated if possible. Mockito lets you write beautiful tests with a clean & simple API. In this example we will learn how to mock a private method. PowerMockito extends Mockito functionality with several new features such as mocking static and private methods and more. Tools and technologies used in this example are Java 1.8, Eclipse Luna 4.4.2

1. Introduction

Mockito is a popular mocking framework which can be used in conjunction with JUnit. Mockito allows us to create and configure mock objects. Using Mockito simplifies the development of tests for classes with external dependencies significantly. We can create the mock objects manually or can use the mocking frameworks like Mockito, EasyMock. jMock etc. Mock frameworks allow us to create mock objects at runtime and define their behavior. The classical example for a mock object is a data provider. In production a real database is used, but for testing a mock object simulates the database and ensures that the test conditions are always the same.

PowerMock provides a class called

```
PowerMockito
```

for creating mock/object/class and initiating verification, and expectations, everything else you can still use Mockito to setup and verify expectation (e.g.

```
times()
```

,

```
anyInt()
```

). All usages require

```
@RunWith(PowerMockRunner.class)
```

and

```
@PrepareForTest
```

annotated at class level.

2. Creating a project

Below are the steps we need to take to create the project.

- Open Eclipse. Go to File=>New=>Java Project. In the 'Project name' enter 'PowerMockSpyExample'.

Figure 1. New Java Project

- Eclipse will create a 'src' folder. Right click on the 'src' folder and choose New=>Package. In the 'Name' text-box enter 'com.javacodegeeks'. Click 'Finish'.

Figure 2. New Package

- Right click on the package and choose New=>Class. Give the class name as PowerMockSpyExample. Click 'Finish'. Eclipse will create a default class with the given name.

Figure 3. New Class

2.1 Dependencies

For this example we need the below mentioned jars:

- junit-4.1.2
- mockito-all-1.10.19
- powermock-mockito-release-full-1.6.4-full
- javassist-3.12.1.GA.jar

These jars can be downloaded from Maven repository. These are the latest (non-beta) versions available as per now. To add these jars in the classpath right click on the project and choose Build Path=>Configure Build Path. The click on the 'Add External JARs' button on the right hand side. Then go to the location where you have downloaded these jars. Then click ok.

Figure 4. Dependencies

3. Code

3.1 Mocking static methods

Follow the below steps to mock the static methods:

- Add

  ```
  @PrepareForTest
  ```

  at class level.

```
@PrepareForTest({StaticClass}.) // StaticClass.class contains static methods
```

- Call

  ```
  PowerMockito.mockStatic()
  ```

  to mock a static class (use

  ```
  PowerMockito.spy(class)
  ```

  to mock a specific method)

3.2 Verify behaviour

Verification of a static method is done in two steps

```
PowerMockito.verifyStatic(); // Step 1
```

```
Static.firstStaticMethod(param);// Step 2
```

You need to call

```
verifyStatic()
```

per method verification.

3.3 Class to Mock

Now we will see how the class which needs to be mocked looks like. We will create a simple class for this example with only one method.

PowerMockSpyExample.java

```java
package com.javacodegeeks;

import java.util.Date;

public class PowerMockSpyExample {

  public String getCurrentDateAsString() {
    return new Date().toGMTString();
  }

}
```

3.4 Test class

Below is the snippet of the test class which will user PowerMock.spy() to mock the desired method:

PowerMockSpyExampleTest.java

```java
package com.javacodegeeks;

import org.junit.Assert;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.mockito.Mockito;
import org.powermock.api.mockito.PowerMockito;
import org.powermock.core.classloader.annotations.PrepareForTest;
import org.powermock.modules.junit4.PowerMockRunner;

/**
* Test class for PowerMockSpyExample
* @author Meraj
*/
@RunWith(PowerMockRunner.class)
@PrepareForTest(PowerMockSpyExample.class)
public class PowerMockSpyExampleTest {

  @Test
  public void testGetCurrentDateAsString() {
    PowerMockSpyExample spy = PowerMockito.spy(new PowerMockSpyExample());

    Mockito.when(spy.getCurrentDateAsString()).thenReturn("Test Date");
    String actual = spy.getCurrentDateAsString();
    Assert.assertEquals("Test Date", actual);
    Mockito.verify(spy, Mockito.times(1)).getCurrentDateAsString();
  }

}
```

4. Download the source file

This was an example of

```
PowerMock.spy().
```

Download
You can download the full source code of this example here: **Power Mock Spy Example**

Tagged with: POWERMOCK

Sign up

KNOWLEDGE BASE

- Courses
- News
- Resources
- Tutorials
- Whitepapers

THE CODE GEEKS NETWORK

- .NET Code Geeks
- Java Code Geeks
- System Code Geeks
- Web Code Geeks

HALL OF FAME

- Android Alert Dialog Example
- Android OnClickListener Example
- How to convert Character to String and a String to Character Array in Java
- Java Inheritance example
- Java write to File Example
- java.io.FileNotFoundException – How to solve File Not Found Exception
- java.lang.arrayindexoutofboundsexception – How to handle Array Index Out Of Bounds Exception
- java.lang.NoClassDefFoundError – How to solve No Class Def Found Error
- JSON Example With Jersey + Jackson
- Spring JdbcTemplate Example

ABOUT JAVA CODE GEEKS

JCGs (Java Code Geeks) is an independent online community focused on crea ultimate Java to Java developers resource center; targeted at the technical ar technical team lead (senior developer), project manager and junior developer JCGs serve the Java, SOA, Agile and Telecom communities with daily news wr domain experts, articles, tutorials, reviews, announcements, code snippets an source projects.

DISCLAIMER

All trademarks and registered trademarks appearing on Java Code Geeks are t property of their respective owners. Java is a trademark or registered tradema Oracle Corporation in the United States and other countries. Examples Java C is not connected to Oracle Corporation and is not sponsored by Oracle Corpor

□ □ □