ABOUT JULEN PARDO

Julen holds his Bachelor's Degree in Computer Engineering from Mondragon Unibertsitatea, in Spain. Currently he Germany, as Software Engineer. He contributes to open source projects with plugins, and he also develops his ow projects. Julen is continuously trying to learn and adopt Software Engineering principles and practices to build be readable and maintainable software.

PowerMock Mock Static Method Example

Posted by: Julen Pardo in PowerMockito June 21st, 2016

In the Mockito Tutorial for Beginners, we did a general introductio mocking framework for JUnit tests. One the things that we did no mocking of static methods. That is because Mockito doesn't allow

To solve this, we will use PowerMock, a framework that extends M functionalities (and other mocking frameworks' also), which allows things, static method mocking.

For this example, we will use:

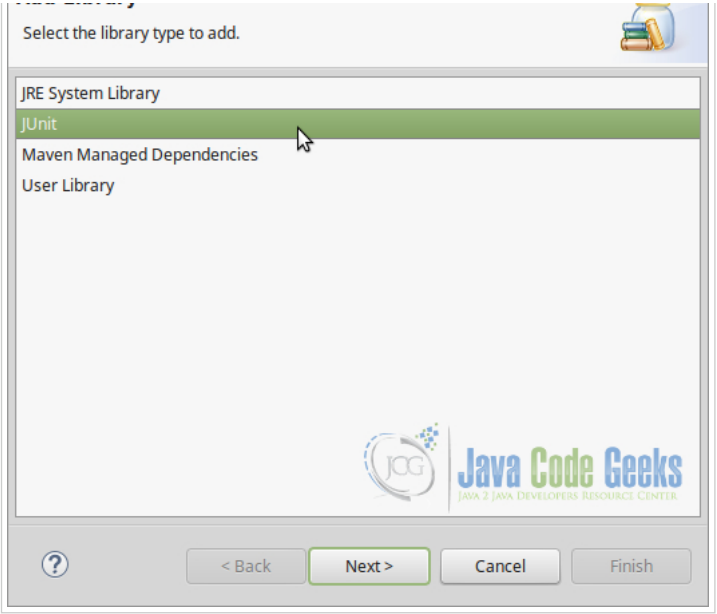
- Java 1.7.0
- Eclipse Mars 2, release 4.5.2.
- JUnit 4.
- PowerMock 1.6.5 for Mockito, and its dependencies.

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

1. Project creation

Go to "File/New/Java Project". You will be asked to enter a name for the project. Then, **press "Next", not "Finish"**.

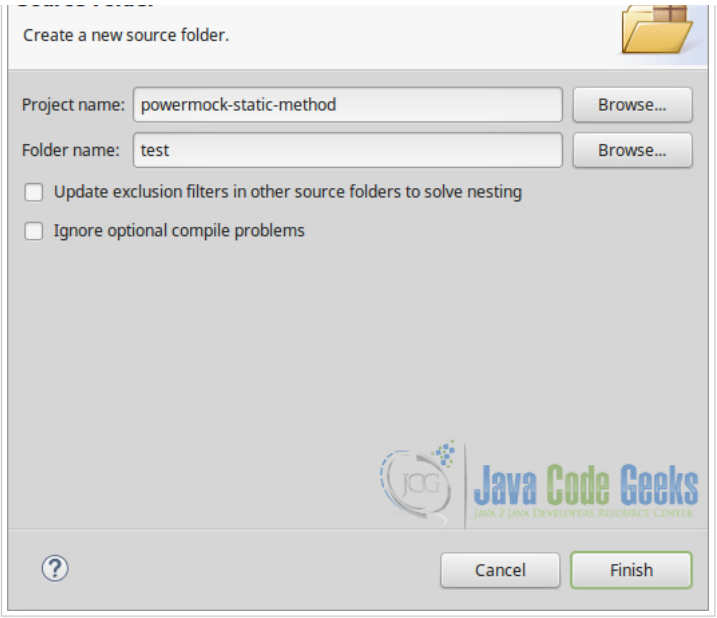
1. Adding libraries to the project



2. Adding JUnit as library

With this, we have added the required dependencies for JUnit testing. You can now finish the project creation.

Now, right click the folder icon in the Package Explorer, and select "New/Source Folder", and enter the name you want to folder.



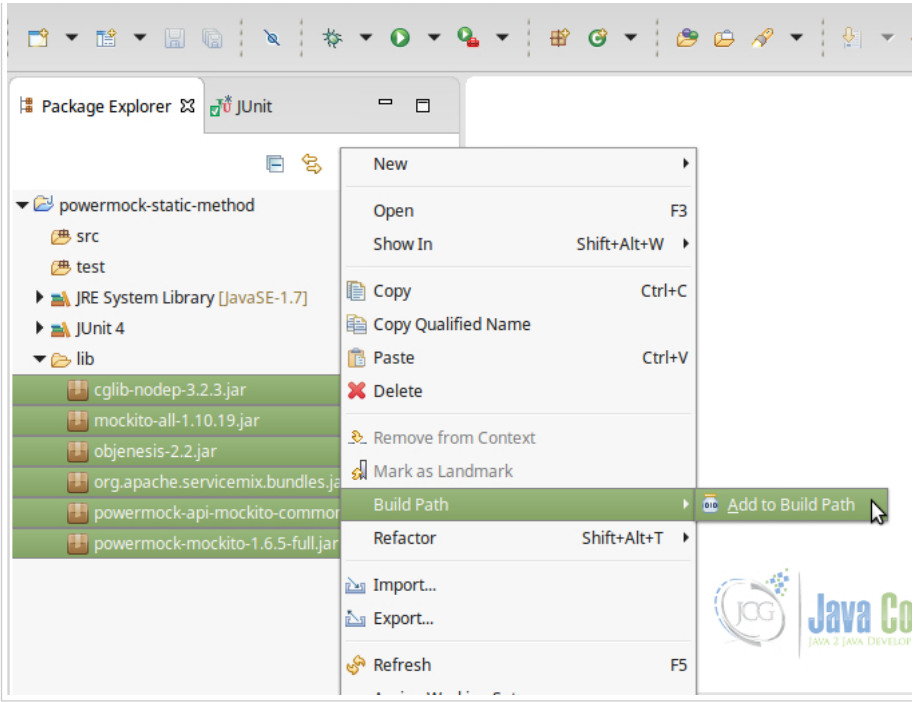
3.Creating folder for tests

1.1. Powermock installation

Apart from JUnit and its dependencies, we need several more libraries.

- Download them:
  - Mockito 1.10.19
  - PowerMock Mockito API
  - PowerMock Mockito (Full)
  - Javassist
  - Objenesis
  - CGLib (nodep)

- Place them inside your working directory, for example, in a lib directory in the directory root.
- Refresh the Package Explorer in Eclipse (F5).
- Now, a new lib directory should be displayed, with all the jar files. Select all of them, right click on them an select "Bui Path" (shown in image below).



4. Adding PowerMock and dependencies to build path

2. Base code

Let's suppose that we have to develop a class for making a query against a database. We could do something similar to tl

DatabaseReader.java

```
01 package com.javacodegeeks.powermock.staticmethod;
02
03 import java.sql.Connection;
04 import java.sql.DriverManager;
05 import java.sql.PreparedStatement;
06 import java.sql.ResultSet;
07 import java.sql.SQLException;
08
09 public class DatabaseReader {
10
11     public static final String CONNECTION = "jdbc:mysql://localhost/testdb";
12
13     public static String getById(int id) throws SQLException {
14         String query = "SELECT * FROM Foo WHERE Id = ?";
15         Connection connection = DriverManager.getConnection(CONNECTION);
16         PreparedStatement preparedStatement = connection.prepareStatement(query);
17         preparedStatement.setInt(1, id);
18         ResultSet resultSet = preparedStatement.executeQuery();
19
20         resultSet.next();
21
```

```
        connection.close();

        return result;
    }
}
```

We won't need a real database (remember that we are mocking).

3. Mocking the method

This is how we would mock our getById static method, using PowerMock:

```
package com.javacodegeeks.powermock.staticmethod;

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.fail;
import static org.mockito.Mockito.when;
import static org.powermock.api.mockito.PowerMockito.mockStatic;
import static org.powermock.api.mockito.PowerMockito.verifyStatic;

import java.sql.SQLException;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.powermock.core.classloader.annotations.PrepareForTest;
import org.powermock.modules.junit4.PowerMockRunner;

@RunWith(PowerMockRunner.class)
@PrepareForTest(DatabaseReader.class)
public class DatabaseReaderTest {

    @Test
    public void testGetById() {
        int inputId = 1;
        String returnValue = "JavaCodeGeeks";

        mockStatic(DatabaseReader.class);

        try {
            when(DatabaseReader.getById(inputId))
                .thenReturn(returnValue);

            String actual = DatabaseReader.getById(inputId);

            verifyStatic();
            assertEquals(returnValue, actual);
        } catch (SQLException e) {
            fail("No exception should be thrown.");
        }
    }

}
```

Which is the main difference with "normal" mocking? Is that **we are specifying that a class' static function will be r**
creating a mock instance and adding behavior to a function. That is done with

```
mockStatic()
```

method and

```
@PrepareForTest
```

annotation, and, then, the behavior is defined as always with

```
when()
```

annotation, and also that we can make the verification of the static method call with

```
verifyStatic()
```

.

Take into account that the following test would work:

DatabaseReaderTest.java

```
@Test
public void testGetById() {
    int inputId = 1;
    String returnValue = "JavaCodeGeeks";

    DatabaseReader databaseReaderMock = Mockito.mock(DatabaseReader.class);
    try {
        when(databaseReaderMock.getById(inputId))
            .thenReturn(returnValue);

        String actual = DatabaseReader.getById(inputId);

        assertEquals(returnValue, actual);
    } catch (SQLException e) {
        fail("No exception should be thrown.");
    }
}

// ...
```

Without the need of PowerMock or any other special mocking technique for the method. **But it would not make any se static method as a object method, since it's supposed to be called as a class method, so the test would not**

3.1. Mocking the database connection

We can go further and **mock what happens inside**

```
getById()
```

method, inside of just adding a predefined behavior to it. For that, we would have to mock the database connecti

```
DriverManager.getConnection()
```

method. With Mockito, we couldn't mock the method in that way. But, as we have seen with PowerMock, we can mock it

DatabaseReaderTest.java

```
// ...

@Test
public void testGetByIdMockDatabase() {
    String query = "SELECT * FROM Foo WHERE Id = ?";
    int inputId = 1;
    String returnValue = "JavaCodeGeeks";

    Connection connectionMock = Mockito.mock(Connection.class);
    PreparedStatement preparedStatementMock = Mockito.mock(PreparedStatement.class)
    ResultSet resultSetMock = Mockito.mock(ResultSet.class);

    mockStatic(DriverManager.class);
    try {
        when(DriverManager.getConnection(DatabaseReader.CONNECTION))
            .thenReturn(connectionMock);
```

```
        when(preparedStatementMock.executeQuery())
            .thenReturn(resultSetMock);

        when(resultSetMock.next())
            .thenReturn(true);

        when(resultSetMock.getString(0))
            .thenReturn(returnValue);

        String actual = DatabaseReader.getById(inputId);

        verify(connectionMock).prepareStatement(query);
        verify(preparedStatementMock).executeQuery();
        verify(resultSetMock).next();
        verify(resultSetMock).getString(0);
        verifyStatic();

        assertEquals(returnValue, actual);
    } catch (SQLException e) {
        fail("No exception should be thrown.");
    }
}

// ...
```

As you can see, apart from mocking the

```
DriverManager.getConnection()
```

static method, we have to create the mocks of the other objects used to make the query, combining PowerMock's static m Mockito's default features; but the concept is the same: **mocking functions that "belong" to class, and not necessa instances**. Without PowerMock, there would not be way to test

```
getById()
```

method with mocks, and a real database would be required.

4. Summary

This tutorial has shown how to mock static methods with PowerMock, a feature that is not available in the Mockito framev **that is specially useful when a method depends on a static method of another class**.

5. Download the Eclipse Project

This was an example of mocking static methods with PowerMock.

Download
You can download the full source code of this example here: **PowerMockMockStaticMethod**

Tagged with: TEST