Java-Success.com

Industrial strength Java Career Companion

Register | Login | Logout | Contact Us

search here … Go

Home | Java FAQs | 600+ Java Q&As | Career | Tutorials | Member | Why?

Can u Debug? | Java 8 ready? | Top X | Productivity Tools | Judging Experience?

Home › Interview › Testing & Profiling/Sampling Java Apps Q&A › Unit Testing Q&A › JUnit Mockito Spring › Part 4: Mockito & PowerMock partially mocking private methods with @Spy

Part 4: Mockito & PowerMock partially mocking private methods with @Spy

Posted on September 14, 2015 by Arulkumaran Kumaraswamipillai

This extends Part 3: Mockito partially mocking with @Spy by making the method "processUser(int id)" as a private method. Mockito framework cannot mock "**private**" methods, hence the "PowerMock" framework is added to the pom.xml file

Step 1: Add "PowerMock" libraries to the pom.xml

```
....
    <properties>
        <project.build.sourceEncoding>UTF-8</pro
        <mockito-all.version>1.9.0</mockito-all.
        <spring.version>3.2.13.RELEASE</spring.v
        <powermock.version>1.4.12</powermock.ver
    </properties>

   <dependencies>
        <dependency>
            <groupId>junit</groupId>
```

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

open all | close all
- Ice Breaker Interview
- Core Java Interview C
- JEE Interview Q&A (3
- Pressed for time? Jav
- SQL, XML, UML, JSC
- Hadoop & BigData Int
- Java Architecture Inte
- Scala Interview Q&As
- Spring, Hibernate, & I
  - Spring (18)
  - Hibernate (13)
  - AngularJS (2)
  - Git & SVN (6)
  - JMeter (2)
  - JSF (2)
  - Maven (3)
- Testing & Profiling/Sa
  - Automation Testing
  - Code Coverage (2)
  - Code Quality (2)
  - jvisualvm profiling (
  - Performance Testir

```
            <artifactId>junit</artifactId>
            <version>4.11</version>
            <scope>test</scope>
        </dependency>

        <!-- Mockito Library -->
        <dependency>
            <groupId>org.mockito</groupId>
            <artifactId>mockito-all</artifactId>
            <version>${mockito-all.version}</ver
            <scope>test</scope>
        </dependency>

        <!-- Power Mock Libraries to mock privat
        <dependency>
            <groupId>org.powermock</groupId>
            <artifactId>powermock-module-junit4<
            <version>${powermock.version}</versi
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.powermock</groupId>
            <artifactId>powermock-api-mockito</a
            <version>${powermock.version}</versi
            <scope>test</scope>
        </dependency>

       //....................
    <dependencies>

```

Step 2: Service and DAO layer interfaces and implementations

Service Layer

```
package com.mytutorial;

public interface SimpleService {
    abstract String processUser(int id);
}

```

```
package com.mytutorial;

import javax.inject.Inject;
import javax.inject.Named;

@Named
public class SimpleServiceImpl implements Simple

    @Inject
    SimpleDao dao;

    public String processUser(int id) {
        return "Hello " +  getUserById(id);
    }

    //made private
```

16 Technical Key Areas

```
    private String getUserById(int id) {
        return dao.getNameById(id);
    }
}

```

DAO Layer

This layer is never reached as the private method "getUserById(int id)" that invokes the dao is mocked.

```
package com.mytutorial;

public interface SimpleDao {
    abstract String getNameById(int id);
}
```

```
package com.mytutorial;

import javax.inject.Named;

@Named
public class SimpleDaoImpl implements SimpleDao

    public String getNameById(int id) {
        if(id == 1) {
            return "John";
        }
        else if (id == 2) {
            return "Peter";
        }
        else {
            throw new IllegalArgumentException()
        }
    }
}
```

Step 3: Now the JUnit class that partially mocks "SimpleServiceImpl" where the method "**processUser(int id)**" is tested and the **private** method "**getUserById(int id)**" is mocked to always return "Sam".

```
package com.mytutorial;

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.annotation
import org.powermock.modules.junit4.PowerMockRun
import org.springframework.test.context.ContextC

```

80+ step by step Java Tutorials

open all | close all

- Setting up Tutorial (6)
- Tutorial - Diagnosis (2
- Akka Tutorial (9)
- Core Java Tutorials (2
- Hadoop & Spark Tuto
- JEE Tutorials (19)
- Scala Tutorials (1)
- Spring & HIbernate Tu
- Tools Tutorials (19)
- Other Tutorials (45)

100+ Java pre-interview coding tests

open all | close all

- Can you write code? (
- ♦ Complete the given
- Converting from A to I
- Designing your classe
- Java Data Structures
- Passing the unit tests
- What is wrong with th
- Writing Code Home A
- Written Test Core Jav
- Written Test JEE (1)

How good are your .....?

open all | close all

- Career Making Know-
- Job Hunting & Resum

```
@RunWith(PowerMockRunner.class)
@ContextConfiguration(classes = { AppConfig.clas
@PrepareForTest(SimpleServiceImpl.class)
public class SimpleTest {

    @Test
    public void testProcessUser() throws Excepti

        SimpleServiceImpl service = PowerMockito

        //invoke partially mocked or spied Simpl
        PowerMockito.doReturn("Sam").when(servic

        String processUser = service.processUser
        Assert.assertEquals("Hello Sam", process
    }
}
```

Note:

1) @RunWith(PowerMockRunner.class)

2) @PrepareForTest(SimpleServiceImpl.class)

3) PowerMockito.spy(new SimpleServiceImpl());

4) PowerMockito.doReturn("Sam").when(service, "getUserById", Mockito.anyInt());

Popular Member Posts

♦ 11 Spring boot interview questions & answers

906 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

816 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

427 views

18 Java scenarios based interview Questions and Answers

409 views

♦ 7 Java debugging interview questions & answers

324 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

312 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

305 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

301 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

251 views

♦ Object equals Vs == and pass by reference Vs value

234 views

0 Like Share Tweet submit reddit 0 G+1 Share

Bio **Latest Posts**

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ Orika Bean Mapping Tutorial   Debugging LinkageErrors in Java ›

Posted in JUnit Mockito Spring, member-paid

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from "Blah blah" to "Wow"? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category ▼

© Disclaimer