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 5: Mockito & PowerMock for mocking static methods

Part 5: Mockito & PowerMock for mocking static methods

Posted on September 17, 2015 by Arulkumaran Kumaraswamipillai

This extends Part 4: Mockito & PowerMock for partially mocking private methods to mock "**static**" methods.

How to use both Spring JUnit and PowerMock runners?

By using the "@PowerMockRunnerDelegate" from the "powermock-module-junit4" jar.

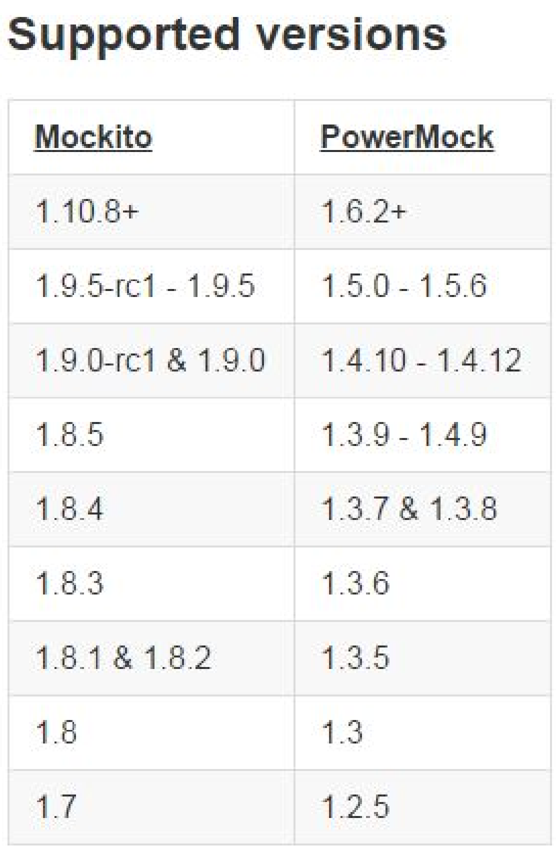
```
@RunWith(PowerMockRunner.class)
@PowerMockRunnerDelegate(SpringJUnit4ClassRunner.
```

So, need to use compatible "Mockito" and "PowerMock".

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

open all | close all

- Ice Breaker Interview
- Core Java Interview (
- 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



Supported versions

Mockito	PowerMock
1.10.8+	1.6.2+
1.9.5-rc1 - 1.9.5	1.5.0 - 1.5.6
1.9.0-rc1 & 1.9.0	1.4.10 - 1.4.12
1.8.5	1.3.9 - 1.4.9
1.8.4	1.3.7 & 1.3.8
1.8.3	1.3.6
1.8.1 & 1.8.2	1.3.5
1.8	1.3
1.7	1.2.5

PowerMock & Mockito version compatibility

Step 1: Add “PowerMock” libraries to the pom.xml

```
....
        <properties>
           <project.build.sourceEncoding>UTF-8</
           <mockito-all.version>1.10.7</mockito-
           <spring.version>3.2.13.RELEASE</sprin
           <powermock.version>1.6.1</powermock.v
        </properties>

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

⊟ Unit Testing Q&A (2
⊞ BDD Testing (4)
⊞ Data Access Uni
⊟ JUnit Mockito Sp
⊟ JUnit Mockito
Spring Con
Unit Testing
Part 1: Unit te
Part 2: Mockit
Part 3: Mockit
Part 4: Mockit
Part 5: Mockit
⊞ Testing Spring T
♦ 5 Java unit tes
JUnit with Hamc
Spring Boot in ui
⊞ Other Interview Q&A f
⊞ ▶ Free Java Interviev

16 Technical Key Areas

open all | close all
⊞ Best Practice (6)
⊞ Coding (26)
⊞ Concurrency (6)
⊞ Design Concepts (7)
⊞ Design Patterns (11)
⊞ Exception Handling (3
⊞ Java Debugging (21)
⊞ Judging Experience Ir
⊞ Low Latency (7)
⊞ Memory Managemen
⊞ Performance (13)
⊞ QoS (8)
⊞ Scalability (4)
⊞ SDLC (6)
⊞ Security (13)
⊞ Transaction Managen

```
        <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 layer interfaces and implementations + utility class that has a static method, and which needs to be mocked with “PowerMock”.

Service Layer

```
package com.mytutorial;

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

```

```
package com.mytutorial;

import javax.inject.Named;

@Named
public class SimpleServiceImpl implements Simple

    public String processUser(int id) {
        return "Hello " +  SimpleUtil.getUserByI
    }

}
```

Utility Class

This static method will be mocked by the “PowerMock”.

```
package com.mytutorial;

public class SimpleUtil {

    public static String getUserById(int id) {
        if (id == 1) {
            return "John";
```

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

```
        } else if (id == 2) {
            return "Peter";
        } else {
            throw new IllegalArgumentException()
        }
    }

}

```

Step 3: Now the JUnit class that mocks “SimpleServiceImpl” where the method “**SimpleUtil.getUserById(int id)**” with the help of PowerMock.

```
package com.mytutorial;

import javax.inject.Inject;

import org.junit.Assert;
import org.junit.Before;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.mockito.Mockito;
import org.mockito.MockitoAnnotations;
import org.powermock.api.mockito.PowerMockito;
import org.powermock.core.classloader.annotation
import org.powermock.modules.junit4.PowerMockRun
import org.powermock.modules.junit4.PowerMockRun
import org.springframework.test.context.ContextC
import org.springframework.test.context.junit4.S

@RunWith(PowerMockRunner.class)
@PowerMockRunnerDelegate(SpringJUnit4ClassRunner
@ContextConfiguration(classes = { AppConfig.clas
@PrepareForTest(SimpleUtil.class)
public class SimpleTest {

    @Inject
    SimpleService service;

    @Before
    public void init() {
        MockitoAnnotations.initMocks(this);
    }

    @Test
    public void testProcessUser() throws Excepti

        PowerMockito.mockStatic(SimpleUtil.class

        //invoke mocked SimpleUtil.getUserById(.
        PowerMockito.when(SimpleUtil.getUserById

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

Note:

1) @RunWith(PowerMockRunner.class)

2) @PowerMockRunnerDelegate(SpringJUnit4ClassRunner.class)

3) @PrepareForTest(SimpleServiceImpl.class)

4) PowerMockito.mockStatic(SimpleUtil.class);

5) PowerMockito.when(SimpleUtil.getUserById(Mockito.anyInt())).thenReturn("Sam");

final methods

It will work if you you add the "final" modfier to the method "SimpleUtil.getUserById(int id)". So, PowerMock is required to mock private, static, and final methods.

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**

‹ Debugging LinkageErrors in Java

Orika Bean Mapping with Spring ›

Posted in JUnit Mockito Spring

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

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.