Java-Success.com

Industrial strength Java Career Companion

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?

Spring Boot in unit testing – 6 things you must know

Posted on October 16, 2015 by Arulkumaran Kumaraswamipillai

Q1. What are some of the libraries required for unit testing Spring based Java applications?
A1.

Spring Test — integration test support for Spring applications.

1. JUnit — for unit testing Java applications.

2. Hamcrest — is a library of matcher objects (aka known as constraints or predicates) allowing assertThat style JUnit assertions.

3. Mockito — is a Java mocking framework to mock objects.

4. PowerMock is an extension to "Mockito" enabling to mock private and static methods.

5. HSQL or H2 DB in memory database to perform DAO or repository integration tests with data.

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

open all | close all

- Ice Breaker Interview
- Core Java Interview Q
- JEE Interview Q&A (3
- Pressed for time? Jav
- SQL, XML, UML, JSO
- Hadoop & BigData Int
- Java Architecture Inte
- Scala Interview Q&As
- Spring, Hibernate, & N
  - 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

6. mockrunner-jms to unit test JMS (i.e. messaging) related logic.

Q2. How does spring boot assist with testing spring applications?

A2. Spring Boot provides a number of useful tools for testing your application. The **spring-boot-starter-test** POM provides Spring Test, JUnit, Hamcrest and Mockito dependencies. There are also useful test utilities in the core spring-boot module under the org.springframework.boot.test package. For example, in the pom.xml, you will have

```
1  <!-- Support for common test dependencies, includ
2  <dependency>
3      <groupId>org.springframework.boot</groupId>
4      <artifactId>spring-boot-starter-test</artifac
5      <scope>test</scope>
6  </dependency>
```

Q3. How will you bootstrap your unit tests in a spring application created with Spring boot?

A3.

```
1  package com.myapp.mom;
2
3  import javax.jms.JMSException;
4  import javax.jms.Message;
5  import javax.jms.ObjectMessage;
6  import javax.jms.Session;
7
8  import org.junit.Ignore;
9  import org.junit.Test;
10 import org.junit.runner.RunWith;
11 import org.slf4j.Logger;
12 import org.slf4j.LoggerFactory;
13 import org.springframework.beans.factory.annotat
14 import org.springframework.boot.test.SpringAppli
15 import org.springframework.jms.core.JmsTemplate;
16 import org.springframework.jms.core.MessageCreat
17 import org.springframework.test.context.junit4.S
18
19 import com.myapp.config.MOMIntegrationTestConfig
20
21 @RunWith(SpringJUnit4ClassRunner.class)
22 @SpringApplicationConfiguration(classes = { MOMI
23 public class JMSMessageSenderIntegrationTest {
24
25     private static final Logger LOG = LoggerFact
26
27     @Autowired
28     private JmsTemplate jmsSenderTemplate;
29
```

16 Technical Key Areas

open all | close all

```
30        @Test
31    public void postMessage() {
32        try {
33            String message = "JMSMessage";
34            MessageCreator wrapMessage = wrapMes
35            LOG.info("Sending message=" + wrapMe
36            jmsSenderTemplate.send(wrapMessage);
37        } catch (Exception e) {
38            LOG.error("Error: " + e);
39        }
40    }
41
42    private MessageCreator wrapMessage(final Str
43        return new MessageCreator() {
44            @Override
45            public Message createMessage(Session
46                ObjectMessage om = session.creat
47                //Message selector properties
48                om.setObjectProperty("messageTyp
49                om.setObjectProperty("originator
50                om.setObjectProperty("category",
51                om.setObjectProperty("env", "EN7
52                return om;
53            }
54        };
55
56    }
57 }
```

Note that "@SpringApplicationConfiguration" is used as opposed to "@ContextConfiguration". The "ContextConfiguration" in "org.springframework.test.context" defines class-level metadata that is used to determine how to load and configure an ApplicationContext for integration tests.

The @ContextConfiguration annotation tells spring how to load and configure the application context. We could also tell spring where it would explicitly find the file, e.g. :

```
1 @ContextConfiguration(locations={myapp/myapp-test
```

OR

```
1 @SpringApplicationConfiguration(classes = { MOMIn
```

Q4. Why use "@SpringApplicationConfiguration" over "@ContextConfiguration" with spring boot?
A4.

1) With the advent of Spring Boot YAML files can be used instead of .properties files. @SpringApplicationConfiguration is required to load the properties from the .yml file.

E.g. application.yml

```
1   env:
2
3   #############################
4   # MQ Settings              #
5   #############################
6
7     mq:
8       hostname: my-mq-host
9       channel: myapp-channel
10      port: 2555
11      queue.manager: MYAPPQUEUEMGR
12      environment: V7
13
```

So, @SpringApplicationConfiguration will load the properties as "env.mq.hostname", "env.mq.channel", etc.

2) @SpringApplicationConfiguration uses uses Spring Boot's "SpringApplicationContextLoader".
SpringApplicationContextLoader assumes that the application is either using 100% XML or 100% Java configuration. So, alternatively, you can do

```
1  @RunWith(SpringJUnit4ClassRunner.class)
2  @ContextConfiguration(classes={TestConfiguration.
3  public class DatabaseIntegrationTests {
4      ...
5  }
```

Q5. What is the purpose of "@WebIntegrationTest" in org.springframework.boot.test?
A5. By annotating your test class (or one of its superclasses) with @WebIntegrationTest, you will be able to test it with HTTP (e.g. using RestTemplate). This is used in web applications to start up and listen on its normal port. if you want to start it on a different port, you can specify the port no, etc as

```
1  @RunWith(SpringJUnit4ClassRunner.class)
```

```
2  @SpringApplicationConfiguration(classes = MyWebAp
3  @WebIntegrationTest({"server.port=8088", "managem
4  public class SomeIntegrationTests {
5
6      // ...
7
8  }
```

Q6. What are some of the utilities provided by the Spring boot?
A6.

1) "ConfigFileApplicationContextInitializer" can be used to load Spring Boot application.properties.

```
1  @ContextConfiguration(classes = MyWebApplicationC
2      initializers = ConfigFileApplicationContextIn
```

2) "OutputCapture" is a JUnit Rule that you can use to capture System.out and System.err output.

```
1  @Rule
2  public OutputCapture capture = new OutputCapture(
3
4  @Test
5  public void testName() throws Exception {
6      System.out.println("Hi John");
7      assertThat(capture.toString(), containsString
8  }
```

3) "TestRestTemplate" is a convenience subclass of Spring's RestTemplate that is useful in integration tests.

```
1
2  //...
3  public class MyWebApplicationTest {
4
5      RestTemplate template = new TestRestTemplate
6
7      @Test
8      public void testMyRequest() throws Exception
9          HttpHeaders headers = template.getForEnt
10         //....
11     }
12 }
```

Popular Member Posts

♦ 11 Spring boot interview questions & answers

907 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	Tweet	▲ submit ▼ reddit	0 G+1	Share
Like				
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**

‹   Injection attack prevention Q&A

   Configuring JMS with Java Config for sending & receiving messages   ›

Posted in member-paid, Unit Testing Q&A

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.