

Industrial strength Java/JEE Career Companion to open more doors

[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](#) > [JEE Interview Q&A](#) > [JMS](#) > Spring JMS with Websphere MQ – configuring and sending messages

Spring JMS with Websphere MQ – configuring and sending messages

Posted on [December 11, 2014](#) by [Arulkumaran Kumaraswamipillai](#) — [No Comments](#) ↓

0
Like
Share

Tweet

0
G+1
Share

Messaging systems are used in enterprise applications for scalability. Here is some sample code that compliments JMS interview questions and answers.

JMS configuration

Step 1: Add the relevant dependency jars to the **pom.xml** file.

```
1
2     <properties>
3         <spring.version>4.2.6.RELEASE</spring.v
```

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](#)
- [JEE Overview \(2\)](#)
- [Web basics \(8\)](#)
- [WebService \(11\)](#)
- [JPA \(2\)](#)
- [JTA \(1\)](#)
- [JDBC \(4\)](#)
- [JMS \(5\)](#)
- [♦ 16 FAQ JMS ir](#)
- [Configuring JMS](#)
- [JMS versus AMC](#)
- [Spring JMS with](#)
- [Spring JMS with](#)
- [JMX \(3\)](#)
- [JNDI and LDAP \(1\)](#)
- [Pressed for time? Jav](#)
- [SQL, XML, UML, JSC](#)
- [Hadoop & BigData Int](#)
- [Java Architecture Inte](#)

```

4      <jms-api.version>2.0.1</jms-api.version>
5      <mq-client.version>8.0.0.0</mq-client.ve
6  </properties>
7  <dependencies>
8      <!-- Spring JMS -->
9      <dependency>
10         <groupId>org.springframework</groupI
11         <artifactId>spring-jms</artifactId>
12         <version>${spring.version}</version>
13     </dependency>
14     <dependency>
15         <groupId>org.springframework</groupI
16         <artifactId>spring-test</artifactId>
17         <version>${spring.version}</version>
18     </dependency>
19
20     <!-- JMS API -->
21     <dependency>
22         <groupId>javax.jms</groupId>
23         <artifactId>javax.jms-api</artifactI
24         <version>${jms-api.version}</version>
25     </dependency>
26
27     <!-- Websphere MQ Implementation -->
28     <dependency>
29         <groupId>com.ibm.mq</groupId>
30         <artifactId>allclient</artifactId>
31         <version>${mq-client.version}</versi
32     </dependency>
33 </dependencies>
34

```

Note that the “com.ibm.mq.allclient.jar” is a third-party i.e. Webspeher MQ jar that comes with IBM Webspehere MQ implementation, and can be manually installed into maven using

```

1
2     mvn install:install-file -Dfile=/home/somedir
3     -DartifactId=allclient -Dversion=8.0.0.0
4

```

You can also install it from within a system path by checking in the third-party library within the project.

```

1
2     <dependency>
3         <groupId>com.ibm.mq</groupId>
4         <artifactId>allclient</artifactId>
5         <version>${mq-client.version}</versio
6         <scope>system</scope>
7         <systemPath>${project.basedir}/lib/co
8     </dependency>
9

```

- [Scala Interview Q&As](#)
- [Spring, Hibernate, & I](#)
- [Testing & Profiling/Sa](#)
- [Other Interview Q&A 1](#)
- [Free Java Interview](#)

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 I](#)
- [Low Latency \(7\)](#)
- [Memory Managemen](#)
- [Performance \(13\)](#)
- [QoS \(8\)](#)
- [Scalability \(4\)](#)
- [SDLC \(6\)](#)
- [Security \(13\)](#)
- [Transaction Managen](#)

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 \(1\)](#)

Step 2: Define the JMS properties

jms/*internalConnection.properties* as shown below

#connection factory properties

```

1  jms.transportType=1 # i.e. TCP
2  jms.hostName=your_host
3  jms.channel=your.channel
4  jms.port=1414
5  jms.queueManager=your.queue.manager
6  jms.sslEnabled=false
7  jms.sslCipherSuite=
8  jms.ssl.keystore.path=
9  jms.ssl.password=
10
11 #destination property
12 my.queueName=my.queue
13
14 my.publishers.count=1
15




```

Step 3: An abstract class that configures the ConnectionFactory.

```











1  package com.myapp.jms;
2
3  import java.util.Properties;
4
5  import javax.jms.ConnectionFactory;
6  import javax.jms.JMSEException;
7
8  import org.slf4j.Logger;
9  import org.slf4j.LoggerFactory;
10 import org.springframework.core.convert.support.
11 import org.springframework.core.convert.support.
12
13 import com.ibm.mq.jms.MQConnectionFactory;
14 import com.ibm.mq.jms.MQQueueConnectionFactory;
15
16 public class AbstractMqJmsConnectionConfig {
17
18     private static final Logger LOG = LoggerFactory
19
20     private final ConfigurableConversionService con
21
22     protected ConnectionFactory createQueueConnecti
23     MQQueueConnectionFactory connectionFactory = n
24
25     setConnectionFactoryProperties(connectionFacto
26
27     return connectionFactory;
28 }
29
30 private void setConnectionFactoryProperties(MQC
31     throws JMSEException {
32
33     connectionFactory.setTransportType(conversionS

```

-  [Spring & Hibernate Ti](#)
-  [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](#)

```

34 connectionFactory.setHostName(properties.getPr
35 connectionFactory.setChannel(properties.getPro
36 connectionFactory.setPort(conversionService.co
37 connectionFactory.setQueueManager(properties.g
38 connectionFactory.setClientID(properties.getPr
39
40 if (conversionService.convert(properties.getPr
41     setSSLSystemProperties(properties);
42     connectionFactory.setSSLCipherSuite(propertie
43 }
44 }
45
46 private void setSSLSystemProperties(Properties
47     String sslkeystoreFullPath = properties.getPro
48     LOG.info("Setting sslkeystoreFullPath : {}", s
49     System.setProperty("javax.net.ssl.keyStore", s
50     System.setProperty("javax.net.ssl.keyStorePass
51 }
52
53 }
54
55

```

Note: The **ConfigurableConversionService** utility class from spring is handy to convert string property values to relevant data types like Integer, Boolean, etc.

Step 4: Define the concrete class that loads the **internalConnection.properties** and has the Spring **@Configuration** annotation for the Spring dependency injection.

```

1 package com.myapp.jms;
2
3 import java.io.IOException;
4 import java.util.Properties;
5
6 import javax.jms.ConnectionFactory;
7 import javax.jms.JMSException;
8
9 import org.springframework.beans.factory.config.
10 import org.springframework.context.annotation.Be
11 import org.springframework.context.annotation.Co
12 import org.springframework.core.io.ClassPathReso
13
14 @Configuration
15 public class InternalJmsConnectionFactoryConfig
16
17
18     @Bean (name="internalJmsConnectionFactory")
19     protected ConnectionFactory createQueueConnecti
20     return createQueueConnectionFactory(internalJM
21 }
22
23
24 private Properties internalJMSProperties() thro
25 PropertiesFactoryBean factory = new Properties
26     factory.setLocation(new ClassPathResource("jms

```

```

27     factory.afterPropertiesSet();
28     return factory.getObject();
29 }
30 }
31

```

Sending JMS messages

Step 5: Configure the *JmsTemplate* with the **@Configurable** Spring annotation

```

1  package com.myapp.jms;
2
3  import java.io.IOException;
4
5  import javax.annotation.Resource;
6  import javax.jms.ConnectionFactory;
7  import javax.jms.JMSException;
8
9  import org.springframework.beans.factory.annotation.Autowired;
10 import org.springframework.context.annotation.Bean;
11 import org.springframework.context.annotation.Configuration;
12 import org.springframework.jms.core.JmsTemplate;
13
14 @Configuration
15 public class MyAppJmsTemplateConfig {
16
17     @Value("${my.queueName}")
18     private String myAppQueueName;
19
20     @Resource (name = "internalJmsConnectionFactory")
21     private ConnectionFactory connectionFactory;
22
23     @Bean(name = "myAppJmsTemplate")
24     public JmsTemplate busniessExceptionJmsTemplate() {
25
26         JmsTemplate jmsTemplate = new JmsTemplate(connectionFactory);
27         jmsTemplate.setExplicitQosEnabled(true);
28         jmsTemplate.setDefaultDestinationName(myAppQueueName);
29         return jmsTemplate;
30     }
31 }
32

```

Step 6: Define the Spring context xml file *META-INF/spring/myApp-applicationContext.xml* file.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4         xmlns:p="http://www.springframework.org/schema/p"
5         xmlns:context="http://www.springframework.org/schema/context"
6         xmlns:aop="http://www.springframework.org/schema/aop"
7         xmlns:jee="http://www.springframework.org/schema/jee"

```

```

8      xmlns:tx="http://www.springframework.org/sch
9      xmlns:util="http://www.springframework.org/s
10     xmlns:task="http://www.springframework.org/s
11     xmlns:mvc="http://www.springframework.org/sc
12     xmlns:jms="http://www.springframework.org/sc
13     xmlns:cache="http://www.springframework.org/
14     xmlns:jdbc="http://www.springframework.org/s
15     xsi:schemaLocation="
16         http://www.springframework.org/schem
17         http://www.springframework.org/schem
18         http://www.springframework.org/schem
19         http://www.springframework.org/schem
20         http://www.springframework.org/schem
21         http://www.springframework.org/schem
22         http://www.springframework.org/schem
23         http://www.springframework.org/schem
24         http://www.springframework.org/schem
25         http://www.springframework.org/schem
26         http://www.springframework.org/schem
27         http://www.springframework.org/schem
28
29     <bean id="placeholderConfig" class="org.springf
30         <property name="ignoreUnresolvablePlaceholde
31         <property name="location" value="classpath:c
32     </bean>
33
34     <context:component-scan base-package="com.myap
35
36 </beans>
37

```

Step 7: Define the message sender class that publishes the message to the default queue that was configured earlier.

```

1  package com.myapp.service.impl;
2
3  import com.google.common.base.Stopwatch;
4  import com.myapp.service.MyAppService;
5
6  import java.util.concurrent.ExecutorService;
7  import java.util.concurrent.Executors;
8  import java.util.concurrent.TimeUnit;
9
10 import javax.annotation.PostConstruct;
11 import javax.annotation.Resource;
12 import javax.annotation.concurrent.ThreadSafe;
13 import javax.jms.JMSException;
14 import javax.jms.Message;
15 import javax.jms.Session;
16
17 import org.slf4j.Logger;
18 import org.slf4j.LoggerFactory;
19 import org.springframework.beans.factory.annotation.Resource;
20 import org.springframework.jms.core.JmsTemplate;
21 import org.springframework.jms.core.MessageCreator;
22 import org.springframework.stereotype.Component;
23 import org.springframework.util.Assert;
24
25 /**
26  * Asynchronously submits a string to the JMS qu

```

```

27  */
28  @Component
29  @ThreadSafe
30  public class MyAppServiceImpl implements MyAppSe
31  {
32
33      private static final Logger LOG = LoggerFact
34
35      @Resource(name = "myAppJmsTemplate")
36      private JmsTemplate jmsTemplate;
37
38      @Value("${my.publishers.count}")
39      int publisherCount;
40
41
42      private ExecutorService pooledSender;
43
44      @PostConstruct
45      void init()
46      {
47          pooledSender = Executors.newFixedThreadP
48      }
49
50
51      @Override
52      public void send(final String msg)
53      {
54          pooledSender.execute(new Runnable()
55          {
56              @Override
57              public void run()
58              {
59                  send(msg, 3);
60              }
61
62              private void send(final String msg,
63              {
64
65                  Stopwatch stopwatch = new Stopwa
66                  try
67                  {
68
69                      jmsTemplate.send(new Message
70                      {
71                          @Override
72                          public Message createMes
73                          {
74                              return session.creat
75                          }
76                      });
77                  }
78                  catch (Exception e)
79                  {
80                      LOG.warn("Unable to send mes
81                  }
82                  LOG.info(" message sent to JMS Q
83              }
84          });
85      }
86  }
87  }
88

```

Step 8: Finally, the ***JUnit*** test class that


```
1 package com.myapp.service.impl;
2
3 import com.myapp.service.MyAppExecutionService;
4
5 import java.util.HashMap;
6 import java.util.Map;
7
8 import javax.annotation.Resource;
9
10 import org.junit.Test;
11 import org.junit.runner.RunWith;
12 import org.slf4j.Logger;
13 import org.slf4j.LoggerFactory;
14 import org.springframework.context.annotation.Primary;
15 import org.springframework.test.context.ContextConfiguration;
16 import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
17
18 @RunWith(SpringJUnit4ClassRunner.class)
19 @ContextConfiguration(locations = "classpath:META-INF/spring/*.xml")
20 @PropertySource({
21     "classpath:jms/internalConnection.properties"
22 })
23 public class MyAppServiceImplSenderIntegrationTest {
24
25     private static final Logger LOG = LoggerFactory.getLogger(MyAppServiceImplSenderIntegrationTest.class);
26
27     @Resource
28     private MyAppExecutionService executionService;
29
30     @Test
31     public void myAppIntegrationTest() {
32         for (int i = 0; i < 1; i++) {
33             final Map<String, String> detailMap = new HashMap<>();
34             detailMap.put("KeyA", "ValueA" + i);
35             detailMap.put("KeyB", "ValueB" + i);
36             executionService.send(detailMap.toString());
37         }
38     }
39 }
40
41
42
43
44
45
```

Popular Posts

♦ [11 Spring boot interview questions & answers](#)

825 views

♦ [Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers](#)

766 views

[18 Java scenarios based interview Questions and Answers](#)

400 views

001A: ♦ 7+ Java integration styles & patterns

interview questions & answers

388 views

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

295 views

♦ 7 Java debugging interview questions & answers

293 views

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

285 views

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

279 views

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

239 views

001B: ♦ Java architecture & design concepts interview questions & answers

201 views

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](#)

◀ 06: ▶ Tell me about yourself? interview Q&A for Java developers

video

Spring JMS with Websphere MQ Listener (Receiver or Subscriber) ▶

Posted in JMS, member-paid

Leave a Reply

Logged in as geethika. [Log out?](#)

Comment

Post Comment

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.