Java Message Service (JMS) Job Interview Questions And Answers

Interview Questions Answers

About Interview Questions Answers

Interview Questions Answers . ORG is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on Java Message Service (JMS) will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit Java Message Service (JMS) Interview Questions And Answers to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in Java Message Service (JMS) category. To ensure quality, each submission is checked by our team, before it becomes live. This Java Message Service (JMS) Interview preparation PDF was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter for latest Jobs and interview preparation guides.
https://twitter.com/InterviewQA

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

Interview Questions Answers.ORG Team
https://InterviewQuestionsAnswers.ORG/
Support@InterviewQuestionsAnswers.ORG

Java Message Service (JMS) Interview Questions And Answers Guide.

Question - 1:

What Is the JMS API?

Ans:

The Java Message Service is a Java API that allows applications to create, send, receive, and read messages. Designed by Sun and several partner companies, the JMS API defines a common set of interfaces and associated semantics that allow programs written in the Java programming language to communicate with other messaging implementations.
The JMS API minimizes the set of concepts a programmer must learn to use messaging products but provides enough features to support sophisticated messaging applications. It also strives to maximize the portability of JMS applications across JMS providers in the same messaging domain.
The JMS API enables communication that is not only loosely coupled but also
* Asynchronous. A JMS provider can deliver messages to a client as they arrive; a client does not have to request messages in order to receive them.
* Reliable. The JMS API can ensure that a message is delivered once and only once. Lower levels of reliability are available for applications that can afford to miss messages or to receive duplicate messages.
The JMS Specification was first published in August 1998. The latest version of the JMS Specification is Version 1.1, which was released in April 2002. You can download a copy of the Specification from the JMS Web site, http://java.sun.com/products/jms/.

View All Answers

Question - 2:

How JMS is different from RPC?

Ans:

In RPC the method invoker waits for the method to finish execution and return the control back to the invoker. Thus it is completely synchronous in nature. While in JMS the message sender just sends the message to the destination and continues it's own processing. The sender does not wait for the receiver to respond. This is asynchronous behavior.

View All Answers

Question - 3:

What type messaging is provided by JMS ?

Ans:

Both synchronous and asynchronous are provided by JMS.

View All Answers

Question - 4:

What is JMS application ?

Ans:

One or more JMS clients that exchange messages.

View All Answers

Question - 5:

What is the difference between Message producer and Message consumer?

Ans:

Messaging systems provide a host of powerful advantages over other conventional distributed computing models. Primarily, they encourage "loose coupling" between message consumers and message producers. There is a high degree of anonymity between producer and consumer: to the message consumer, it doesn't matter who produced the message, where the producer lives on the network, or when the message was produced.

View All Answers

Question - 6:

What is the difference between durable and non-durable subscriptions?

Ans:

Point-To-Point (PTP). This model allows exchanging messages via queues created for some purposes. A client can send and receive messages from one or several queues. PTP model is easier than pub/sub model.
A durable subscription gives a subscriber the freedom of receiving all messages from a topic, whereas a non-durable subscription doesn't make any guarantees about messages sent by others when a client was disconnected from a topic.

Question - 7:

What is the use of JMS? In which situations we are using JMS? Can we send message from one server to another server using JMS?

Ans:

JMS is the ideal high-performance messaging platform for intrabusiness messaging, with full programmatic control over quality of service and delivery options.

Question - 8:

What is JMS session?

Ans:

A single-threaded context for sending and receiving JMS messages. A JMS session can be nontransacted, locally transacted, or participating in a distributed transaction.

Question - 9:

How does a typical client perform the communication? Queue

Ans:

1. Use JNDI to locate administrative objects.
2. Locate a single ConnectionFactory object.
3. Locate one or more Destination objects.
4. Use the ConnectionFactory to create a JMS Connection.
5. Use the Connection to create one or more Session(s).
6. Use a Session and the Destinations to create the MessageProducers and MessageConsumers needed.
7. Perform your communication.

Question - 10:

What is asynchronous messaging? Queue

Ans:

Asynchronous messaging involves a client that does not wait for a message from the server. An event is used to trigger a message from a server. So even if the client is down , the messaging will complete successfully.

Question - 11:

What is synchronous messaging? Queue

Ans:

Synchronous messaging involves a client that waits for the server to respond to a message. So if one end is down the entire communication will fail.

Question - 12:

Does JMS specification define transactions? Queue

Ans:

JMS specification defines a transaction mechanisms allowing clients to send and receive groups of logically bounded messages as a single unit of information. A Session may be marked as transacted. It means that all messages sent in a session are considered as parts of a transaction. A set of messages can be committed (commit() method) or rolled back (rollback() method). If a provider supports distributed transactions, it's recommended to use XAResource API.

Question - 13:

What is the difference between Java Mail and JMS Queue?

Ans:

JMS is the ideal high-performance messaging platform for intrabusiness messaging, with full programmatic control over quality of service and delivery options.
JavaMail provides lowest common denominator, slow, but human-readable messaging using infrastructure already available on virtually every computing platform.

Question - 14:

What is the Role of the JMS Provider?

Ans:

The JMS provider handles security of the messages, data conversion and the client triggering. The JMS provider specifies the level of encryption and the security level of the message, the best data type for the non-JMS client.

View All Answers

Question - 15:

What Is Messaging?

Ans:

Messaging is a method of communication between software components or applications. A messaging system is a peer-to-peer facility: A messaging client can send messages to, and receive messages from, any other client. Each client connects to a messaging agent that provides facilities for creating, sending, receiving, and reading messages.
Messaging enables distributed communication that is loosely coupled. A component sends a message to a destination, and the recipient can retrieve the message from the destination. However, the sender and the receiver do not have to be available at the same time in order to communicate. In fact, the sender does not need to know anything about the receiver; nor does the receiver need to know anything about the sender. The sender and the receiver need to know only what message format and what destination to use. In this respect, messaging differs from tightly coupled technologies, such as Remote Method Invocation (RMI), which require an application to know a remote application's methods.
Messaging also differs from electronic mail (e-mail), which is a method of communication between people or between software applications and people. Messaging is used for communication between software applications or software components.
Messaging is a mechanism by which data can be passed from one application to another application.

View All Answers

Question - 16:

What is the main parts of JMS applications?

Ans:

The main parts of JMS applications are:
--ConnectionFactory and Destination
--Connection
--Session
--MessageProducer
--MessageConsumer
--Message

View All Answers

Question - 17:

What are the different parts of a JMS message ?

Ans:

A JMS message contains three parts. a header, an optional properties and an optional body.

View All Answers

Question - 18:

Which models are supported by JMS? Please, explain them.

Ans:

Publish/subscribe (pub/sub). This model allows a client (publisher) to send messages to a JMS topic. These messages are retrieved by other clients (subscribers) (it may happen so that a topic has no subscribers) asynchronously. Pub/sub model requires a broker distributing messages to different consumers.

View All Answers

Question - 19:

What is publish/subscribe messaging?

Ans:

With publish/subscribe message passing the sending application/client establishes a named topic in the JMS broker/server and publishes messages to this queue. The receiving clients register (specifically, subscribe) via the broker to messages by topic; every subscriber to a topic receives each message published to that topic. There is a one-to-many relationship between the publishing client and the subscribing clients.

View All Answers

Question - 20:

What is JMS administered object?

Ans:

A preconfigured JMS object (a resource manager connection factory or a destination) created by an administrator for the use of JMS clients and placed in a JNDI namespace

View All Answers

Question - 21:

What is the publish-and-subscribe model in JMS?

Ans:

A publish-subscribe model is based on the message topic concept: Publishers send messages in a topic, and all subscribers of the given topic receive these messages.

Question - 22:

What are the advantages of JMS?

Ans:

One of the principal advantages of JMS messaging is that it's asynchronous. Thus not all the pieces need to be up all the time for the application to function as a whole.

Question - 23:

What is the point-to-point model in JMS?

Ans:

A point-to-point model is based on the concept of a message queue: Senders send messages into the queue, and the receiver reads messages from this queue. In the point-to-point model, several receivers can exist, attached to the same queue. However, (Message Oriented Middleware)MOM will deliver the message only to one of them. To which depends on the MOM implementation.

Question - 24:

How may messaging models do JMS provide for and what are they?

Ans:

JMS provides for two messaging models, publish-and-subscribe and point-to-point queuing.

Question - 25:

What type messaging is provided by JMS?

Ans:

Both synchronous and asynchronous.

Question - 26:

Must I place all my class files in the WEB-INF folder and all JSP's outside?

Ans:

The class files should place into WEB-INF/classes folder and the JSP files should place within a separate folder.

Question - 27:

What is JMS?

Ans:

Java Message Service: An interface implemented by most J2EE containers to provide point-to-point queueing and topic (publish/subscribe) behavior. JMS is frequently used by EJB's that need to start another process asynchronously.

For example, instead of sending an email directly from an Enterprise JavaBean, the bean may choose to put the message onto a JMS queue to be handled by a Message-Driven Bean (another type of EJB) or another system in the enterprise. This technique allows the EJB to return to handling requests immediately instead of waiting for a potentially lengthy process to complete.

Java Programing Most Popular & Related Interview Guides

1 : **Core Java Interview Questions and Answers.**

2 : **Hibernate Interview Questions and Answers.**

3 : **IBM WebSphere Interview Questions and Answers.**

4 : **Advanced Java Interview Questions and Answers.**

5 : **Spring Framework Interview Questions and Answers.**

6 : **Full Stack Developer (Java) Interview Questions and Answers.**

7 : **JSF Interview Questions and Answers.**

8 : **JDBC Interview Questions and Answers.**

9 : **Java Swing Programming Interview Questions and Answers.**

10 : **Java JSP Programming Interview Questions and Answers.**

Follow us on FaceBook
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter
https://twitter.com/InterviewQA

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team
https://InterviewQuestionsAnswers.ORG/
support@InterviewQuestionsAnswers.ORG