

Java EJB Programming Job Interview Questions And Answers



Interview Questions Answers

<https://interviewquestionsanswers.org/>

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 EJB Programming 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 EJB Programming 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 EJB Programming category. To ensure quality, each submission is checked by our team, before it becomes live. This [Java EJB Programming 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 EJB Programming Interview Questions And Answers Guide.

Question - 1:

What is the difference between session and entity beans? When should I use one or the other?

Ans:

An entity bean represents persistent global data from the database; a session bean represents transient user-specific data that will die when the user disconnects (ends his session). Generally, the session beans implement business methods (e.g. Bank.transferFunds) that call entity beans (e.g. Account.deposit, Account.withdraw)

[View All Answers](#)

Question - 2:

What is the default transaction attribute for an EJB?

Ans:

There is no default transaction attribute for an EJB. Section 11.5 of EJB v1.1 spec says that the deployer must specify a value for the transaction attribute for those methods having container managed transaction. In WebLogic, the default transaction attribute for EJB is SUPPORTS.

[View All Answers](#)

Question - 3:

Is it possible to stop the execution of a method before completion in a SessionBean?

Ans:

Stopping the execution of a method inside a Session Bean is not possible without writing code inside the Session Bean. This is because you are not allowed to access Threads inside an EJB.

[View All Answers](#)

Question - 4:

Is it legal to have static initializer blocks in EJB?

Ans:

Although technically it is legal, static initializer blocks are used to execute some piece of code before executing any constructor or method while instantiating a class. Static initializer blocks are also typically used to initialize static fields - which may be illegal in EJB if they are read/write - In EJB this can be achieved by including the code in either the ejbCreate(), setSessionContext() or setEntityContext() methods.

[View All Answers](#)

Question - 5:

Is it possible for an EJB client to marshal an object of class java.lang.Class to an EJB?

Ans:

Technically yes, spec. compliant NO! - The enterprise bean must not attempt to query a class to obtain information about the declared members that are not otherwise accessible to the enterprise bean because of the security rules of the Java language.

[View All Answers](#)

Question - 6:

What is an EJB Context?

Ans:

EJBContext is an interface that is implemented by the container, and it is also a part of the bean-container contract. Entity beans use a subclass of EJBContext called EntityContext. Session beans use a subclass called SessionContext. These EJBContext objects provide the bean class with information about its container, the client using the bean and the bean itself. They also provide other functions. See the API docs and the spec for more details.

[View All Answers](#)



Question - 7:

How can I call one EJB from inside of another EJB?

Ans:

EJBs can be clients of other EJBs. It just works. Use JNDI to locate the Home Interface of the other bean, then acquire an instance reference, and so forth.

[View All Answers](#)

Question - 8:

What is the difference between Message Driven Beans and Stateless Session beans?

Ans:

In several ways, the dynamic creation and allocation of message-driven bean instances mimics the behavior of stateless session EJB instances, which exist only for the duration of a particular method call. However, message-driven beans are different from stateless session EJBs (and other types of EJBs) in several significant ways: Message-driven beans process multiple JMS messages asynchronously, rather than processing a serialized sequence of method calls. Message-driven beans have no home or remote interface, and therefore cannot be directly accessed by internal or external clients. Clients interact with message-driven beans only indirectly, by sending a message to a JMS Queue or Topic.

Only the container directly interacts with a message-driven bean by creating bean instances and passing JMS messages to those instances as necessary. The Container maintains the entire lifecycle of a message-driven bean; instances cannot be created or removed as a result of client requests or other API calls.

[View All Answers](#)

Question - 9:

What happens if remove() is never invoked on a session bean?

Ans:

In case of a stateless session bean it may not matter if we call or not as in both cases nothing is done. The number of beans in cache is managed by the container.

In case of stateful session bean, the bean may be kept in cache till either the session times out, in which case the bean is removed or when there is a requirement for memory in which case the data is cached and the bean is sent to free pool.

[View All Answers](#)

Question - 10:

What are the special design care that must be taken when you work with local interfaces?

Ans:

It is important to understand that the calling semantics of local interfaces are different from those of remote interfaces. For example, remote interfaces pass parameters using call-by-value semantics, while local interfaces use call-by-reference.

This means that in order to use local interfaces safely, application developers need to carefully consider potential deployment scenarios up front, then decide which interfaces can be local and which remote, and finally, develop the application code with these choices in mind. While EJB 2.0 local interfaces are extremely useful in some situations, the long-term costs of these choices, especially when changing requirements and component reuse are taken into account, need to be factored into the design decision.

[View All Answers](#)

Question - 11:

Brief description about local interfaces?

Ans:

EEJB was originally designed around remote invocation using the Java Remote Method Invocation (RMI) mechanism, and later extended to support to standard CORBA transport for these calls using RMI/IIOP. This design allowed for maximum flexibility in developing applications without consideration for the deployment scenario, and was a strong feature in support of a goal of component reuse in J2EE. Many developers are using EJBs locally, that is, some or all of their EJB calls are between beans in a single container. With this feedback in mind, the EJB 2.0 expert group has created a local interface mechanism. The local interface may be defined for a bean during development, to allow streamlined calls to the bean if a caller is in the same container. This does not involve the overhead involved with RMI like marshalling etc.

This facility will thus improve the performance of applications in which co-location is planned. Local interfaces also provide the foundation for container-managed relationships among entity beans with container-managed persistence.

[View All Answers](#)

Question - 12:

What is EJB QL?

Ans:

EJB QL is a Query Language provided for navigation across a network of enterprise beans and dependent objects defined by means of container managed persistence. EJB QL is introduced in the EJB 2.0 specification. The EJB QL query language defines finder methods for entity beans with container managed persistence and is portable across containers and persistence managers. EJB QL is used for queries of two types of finder methods: Finder methods that are defined in the home interface of an entity bean and which return entity objects. Select methods, which are not exposed to the client, but which are used by the Bean Provider to select persistent values that are maintained by the Persistence Manager or to select entity objects that are related to the entity bean on which the query is defined.

[View All Answers](#)

Question - 13:

What is the advantage of using Entity bean for database operations, over directly using JDBC API to do database operations? When would I use one over the other?

Ans:

Entity Beans actually represents the data in a database. It is not that Entity Beans replaces JDBC API. There are two types of Entity Beans Container Managed and Bean Managed. In Container Managed Entity Bean - Whenever the instance of the bean is created the container automatically retrieves the data from the DB/Persistence storage and assigns to the object variables in bean for user to manipulate or use them. For this the developer needs to map the fields in the database to the variables in deployment descriptor files (which varies for each vendor). In the Bean Managed Entity Bean - The developer has to specifically make connection,



retrieve values, assign them to the objects in the `ejbLoad()` which will be called by the container when it instantiates a bean object. Similarly in the `ejbStore()` the container saves the object values back to the persistence storage. `ejbLoad` and `ejbStore` are callback methods and can be only invoked by the container. Apart from this, when you use Entity beans you don't need to worry about database transaction handling, database connection pooling etc. which are taken care by the ejb container.

[View All Answers](#)

Question - 14:

Can you control when passivation occurs?

Ans:

The developer, according to the specification, cannot directly control when passivation occurs. Although for Stateful Session Beans, the container cannot passivate an instance that is inside a transaction. So using transactions can be a strategy to control passivation. The `ejbPassivate()` method is called during passivation, so the developer has control over what to do during this exercise and can implement the required optimized logic. Some EJB containers, such as BEA WebLogic, provide the ability to tune the container to minimize passivation calls.

Taken from the WebLogic 6.0 DTD -The passivation-strategy can be either default or transaction. With the default setting the container will attempt to keep a working set of beans in the cache. With the transaction setting, the container will passivate the bean after every transaction (or method call for a non-transactional invocation).

[View All Answers](#)

Question - 15:

Can the primary key in the entity bean be a Java primitive type such as int?

Ans:

The primary key can't be a primitive type. Use the primitive wrapper classes, instead. For example, you can use `java.lang.Integer` as the primary key class, but not `int` (it has to be a class, not a primitive).

[View All Answers](#)

Question - 16:

The EJB container implements the `EJBHome` and `EJBObject` classes. For every request from a unique client, does the container create a separate instance of the generated `EJBHome` and `EJBObject` classes?

Ans:

The EJB container maintains an instance pool. The container uses these instances for the EJB Home reference irrespective of the client request, while referring the EJB Object classes the container creates a separate instance for each client request. The instance pool maintenance is up to the implementation of the container. If the container provides one, it is available otherwise it is not mandatory for the provider to implement it. Having said that, yes most of the container providers implement the pooling functionality to increase the performance of the application server. The way it is implemented is, again, up to the implementer.

[View All Answers](#)

Question - 17:

Is it possible to share an `HttpSession` between a JSP and EJB? What happens when I change a value in the `HttpSession` from inside an EJB?

Ans:

You can pass the `HttpSession` as parameter to an EJB method, only if all objects in session are serializable. This has to be considered as passed-by-value, that means that it's read-only in the EJB. If anything is altered from inside the EJB, it won't be reflected back to the `HttpSession` of the Servlet Container. The pass-by-reference can be used between EJBs Remote Interfaces, as they are remote references.

While it is possible to pass an `HttpSession` as a parameter to an EJB object, it is considered to be bad practice in terms of object-oriented design. This is because you are creating an unnecessary coupling between back-end objects (EJBs) and front-end objects (`HttpSession`). Create a higher-level of abstraction for your EJBs API. Rather than passing the whole, fat, `HttpSession` (which carries with it a bunch of http semantics), create a class that acts as a value object (or structure) that holds all the data you need to pass back and forth between front-end/back-end. Consider the case where your EJB needs to support a non HTTP-based client. This higher level of abstraction will be flexible enough to support it.

[View All Answers](#)

Question - 18:

How EJB Invocation happens?

Ans:

Retrieve Home Object reference from Naming Service via JNDI. Return Home Object reference to the client. Create a new EJB Object through Home Object interface. Create EJB Object from the Ejb Object. Return EJB Object reference to the client. Invoke business method using EJB Object reference. Delegate request to Bean (Enterprise Bean).

[View All Answers](#)

Question - 19:

What is Entity Bean and Session Bean ?

Ans:

Entity Bean is a Java class which implements an Enterprise Bean interface and provides the implementation of the business methods. There are two types: Container Managed Persistence(CMP) and Bean-Managed Persistence(BMP).

Session Bean is used to represent a workflow on behalf of a client. There are two types: Stateless and Stateful. Stateless bean is the simplest bean. It doesn't maintain any conversational state with clients between method invocations. Stateful bean maintains state between invocations.

[View All Answers](#)

Question - 20:

What is message-driven bean?

**Ans:**

A message-driven bean combines features of a session bean and a Java Message Service (JMS) message listener, allowing a business component to receive JMS. A message-driven bean enables asynchronous clients to access the business logic in the EJB tier.

[View All Answers](#)

Question - 21:

How many enterprise beans?

Ans:

There are three kinds of enterprise beans:

1. session beans,
2. entity beans, and
3. message-driven beans.

[View All Answers](#)

Question - 22:

What are the key benefits of the EJB technology?

Ans:

- * Rapid application development
- * Broad industry adoption
- * Application portability
- * Protection of IT investment

[View All Answers](#)

Question - 23:

What are the key features of the EJB technology?

Ans:

- * EJB components are server-side components written entirely in the Java programming language
- * EJB components contain business logic only - no system-level programming & services, such as transactions, security, life-cycle, threading, persistence, etc. are automatically managed for the EJB component by the EJB server.
- * EJB architecture is inherently transactional, distributed, portable multi-tier, scalable and secure.
- * EJB components are fully portable across any EJB server and any OS.
- * EJB architecture is wire-protocol neutral--any protocol can be utilized like IIOP, JRMP, HTTP, DCOM, etc.

[View All Answers](#)

Question - 24:

What is the difference between EJB and Java beans?

Ans:

EJB is a specification for J2EE server, not a product; Java beans may be a graphical component in IDE.

[View All Answers](#)

Question - 25:

What is EJB role in J2EE?

Ans:

EJB technology is the core of J2EE. It enables developers to write reusable and portable server-side business logic for the J2EE platform.

[View All Answers](#)

Question - 26:

What is session Facade in EJB?

Ans:

Session Facade is a design pattern to access the Entity bean through local interface than accessing directly. It increases the performance over the network. In this case we call session bean which on turn call entity bean.

[View All Answers](#)

Question - 27:

What is EJB?

Ans:

EJB stands for Enterprise JavaBean and is a widely-adopted server side component architecture for J2EE. It enables rapid development of mission-critical application that are versatile, reusable and portable across middleware while protecting IT investment and preventing vendor lock-in.

[View All Answers](#)

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](https://InterviewQuestionsAnswers.ORG/support@InterviewQuestionsAnswers.ORG)