**What is the difference between EJB3.0 and EJB2.0?**

EJB3.0 is the latest version of EJB(Enterprise JavaBeans). EJB2.0 is the second major release of EJB that happened before the release of EJB2.0. EJB3.0 has introduced many new features of enhancements when compared to EJB2.0. The major differences between EJB3.0 and EJB2.0 are:

* Elimination of home interface gives simpler lookup process in EJB3.0 than in EJB2.0.
* Employment of annotations makes EJB3.0 a latest technology while the EJB2.0 has home and remote interfaces instead of annotations.
* EJB2.0 deploys entity beans for accessing database. EJB3.0 has Java Persistence API to access data which is generalized to address issues of portability.
* EJB3.0 performs better because it uses POJOs along with metadata annotations, a new introduction. The POJOs are faster than XMLDescriptor and JNDI deployed by EJB2.0 for reference of objects. EJB2.0 does not implement POJO.
* EJB2.0 is heavy when it comes to writing home and remote interfaces. EJB3.0 has no restriction on using any of the standard interfaces. It is a well configured and simple POJO which doesn’t need to execute container callback methods such as ejbActive, ejbStore, etc which are used by EJB2.0.
* In EJB3.0, entity bean can be effortlessly converted to DAO and vice versa. This is not possible in EJB2.0.

**What do you mean by EJB?**

Ans: Most of the time this EJB interview questions is the first questions asked to interviewee, mostly to check how he responds and how its is different than traditional Java beans. Enterprise java bean is a server side component which runs on application server or we call container, developed for the purpose of distributed and enterprise level application .container will provide support for system level services like [Transaction Management](http://javarevisited.blogspot.com/2011/11/database-transaction-tutorial-example.html), security which make developer task easy and he can focus on business logic.

What are the different types of EJB?

This is another beginner level EJB interview questions mostly asked in telephonic interviews and appeared in 2 to 3 years experience level interviews. Mainly three types of EJB, Entity Bean, Session Bean and Message Driven Bean(MDB).

Types of EJB:

Entity Bean: it represents an entity which is mapped with database or we can say it makes OR object Relational mapping with Database. Entity bean typically represent table in RDBMS and each instance represent row in the table.

 Two types of entity bean:

          CMP Entity bean: Container managed entity bean its responsibility of container to manage the bean persistence behavior.

          BMP Entity bean: Programmer manage the bean persistence behavior.

Session bean: Session bean is responsible for developing business logic it makes the client server relationship so session beans exist till there is a session exist between client and server, it doesn’t contain persistent business concept.

 Types of session bean

          Stateless session bean: when there is not need to maintain state of a particular client stateless session bean is used .They alive for short period of time.

       For example if we are validating the credit card we can use stateless session bean.

          Stateful session bean: stateful session bean maintain the conversational state of client over the series of method call before the bean instance goes to passive state conversational state is saved to persistence area like Hard disk and again when same client send a request and bean instance come into the active state it will come out from hard disk to main memory.

      For Example when we do online banking transaction, online reservation we use stateful session bean

Message Driven Beans: these beans are work as a listener for messaging services like JMS.

Question 3: Explain the life cycle method of EJB?

One for EJB interview questions which is asked in Junior level interviews. I have not seen this EJB question appeared on more senior level interviews but its good to remember life cycle as here you have chance to show how much EJB you know in depth.

Ans: Life Cycle of Entity Bean:

 First stage is Does Not Exist Stage then Container creates the instance of EJB and  call SetEntityContext() method  which will set all entity context to bean and now it will become available on pool ,to get a particular identity of an EJB object it has to move from Pooled stage to ready stage which is done by calling the create() method which in turns call ejbCreate() and ejbPostCreate() method .

There is another way by which directly entity bean can move to pooled stage to ready stage that’s is call ejbActivate() method. now we are ready to invoke the business method of entity bean .After completion of business method if we want to move again in pooled stage from ready stage we can call remove() method which in turns call ejbRemove() or directly call ejbPassivate () method. At the end container remove the instance of EJBfrom pool and call unSetEntityContext().

Life Cycle of Stateful Session Bean :

Stateful session beans life cycle starts when  client call create() method.The container create the instance of session bean and call setSessionContext() and ejbCreate() method

Now the stateful session bean is ready to serve the client request after serving the request if it is not used after a long time container can move this bean to passive stage by calling the ejbPassivate() method. Similarly when bean is in passive stage and client invoke the business method the container call ejbActivate() method to move bean from passive stage to active or ready stage.

At the end of life cycle client  call remove() method and container will call ejbRemove() method and bean is ready for garbage collection.

Life Cycle of Stateless session bean :

Stales session bean has short life cycle it can have two stage does not exist and ready stage. ejb container create the instance of stateless session bean and call setSessionContext () and ejbCreate() method.Now the bean is ready to invoke business method on this.it will not maintain the state so remove () method is been called after completion of business method which in turns  call ejbRemove () and now its ready for  garbage collection.

Life cycle of Message Driven bean:

MDBs have same life cycle like stateless session bean. setMessageDrivenContext() method and ejbCreate() method is called by container to create the instance of MDB.now its ready to receive message .and at the end of lifecycle client will call remove () method

which in turns  call ejbRemove () and now its ready for  garbage collection.

Question 4 : Can we have static initializer Block in EJB.

This is one of the tricky EJB interview questions which makes you think and some time left you stunned with feeling like “ Ah , I haven’t thought about it”.

Ans .Purpose of Static initialize block is to initialize some static fields before any execution of constructor or method or we can say creation of object. According to EJB Spec it’s a violation if static field are used and they are non final .

“EJB Spec”

Enterprise bean are not allowed to read or write the non final fields.

But technically, from the point of java its correct but if in EJB we use static initializer block to initialize static field then because EJB components are used in distributed environment mean to say if  single JVM then it will be ok but if run on different JVM then it will be a problem if we change or update the value in on environment then only the instance running on same JVM have new value .that’s why static blocks are avoided and also all static field should be final.

And same thing we can achieve in ejbCreate(), setSessionContext() or setEntityContext() methods.

Question 5: Is threading possible in EJB?

Another tricky EJB interview question, I love to ask this kind of question because it shows real knowledge of EJB which is important to avoid mistakes while writing enterprise Java application.

Ans: No, Not possible because EJBs are created and managed by container and if in ejbs we allow threading containers life cycle methods will be interrupted by us and also the purpose of EJB is to perform business logic not to controlling a system or implementation level functioning so container will manage the thread and developer can concentrate on business logic.

Question 6: What is connection pooling feature of EJB container?

This EJB interview question is neither tough nor easy, as most of Java programmer familiar with concept of connection pooling that makes it easy but how EJB container does that is something you want to hear from interviewee.

Ans: Connection pooling is one of the Advance feature of container which enhanced our application performance .Using connection pooling mechanism client are not required to create every time connection object to interact with database .Here in pool objects are already available Whenever a client request for a database connection then an instance is picked from the connection pool to get an access to database and when user complete with his work instance is returned to connection pool. There is a limit specified by App server administrator for the availability of number of connections and beyond a specified limit a predefined number increases numbers of connection pool instances. When demand goes back to normal then access amount of connection pool instances are removed.

Question 7: what is Session facade?

This is the most popular interview questions asked in EJB in connection of Session Bean. Session facade not only touch concept of EJB but also it checks design patterns knowledge of interviewee.

Ans: Session Façade is one of the design pattern which is used in EJB component in this pattern session beans are used as a wrapper classes for entity bean and using session bean we interact with Entity bean .it will give clean approach towards client access to bean components and also reduce network calls so as to make the whole system of high performance.

Question 8:What are the transaction Attribute ?

This kind of EJB questions mostly asked in telephonic interviews but believe me its not easy to answer if you have not really used transaction management feature of EJB. Its easy to mug all these transaction attribute and spill in interview but once interviewer asked cross question you will most likely cripple.

Ans: Transaction is group of operation or work which should be performed either completely or none to maintain the data integrity. All transaction must have ACID property (atomicity, consistency, integrity,durability) so transaction can be said completed if we commit on successful execution and rollback on unsuccessful execution.

There are two ways of transaction management.

          Declarative Transaction Mang.

          Programmatic Transaction Mang.

Now we see what the transactions Attribute are

Transaction Attribute conveys to the container the intended transactional behavior of the associated EJB component's method.

Required: if required attribute is associated with Method then new Transaction context may or may not be created, means if method is already associated with transaction context then no new transaction context should be created.

Requires New: if Requires New attribute is associated with Method then always new Transaction context may be created.

NotSupported: if Method is Associated with this Attribute then method is  a not a part of transaction.

Supported: if a Method is Associated with this transaction Attribute then method will act as Not supported if calling component is not associated with transaction and if calling component is associated with transaction then act as a required Attribute.

Mandatory: if a method is Associated with this attribute then always be called from calling component transaction context.

Never: if a method is associated with this attribute then never be called from calling component transaction context.

The default transaction value for EJB is **Support.**

**Question 9: Difference Between ejbStore() and ejbLoad()?**

One more popular EJB interview question, not seen though it for some time but still an important questions on EJB to prepare.

Ans: Both of these methods are used in context of entity bean for the purpose of synchronizing the instance variable to corresponding value store in the database.ejbLoad method load or refresh the instance variable from the database and ejbStore method writes variables value back to the database.

**Question 10**:**What is the difference between JNDI context, Initial context, session context and E**

**jb context.**

JNDI questions most likely asked in any EJB interview so be prepare for that as well.

**Ans:**

**JNDI context**: provides mechanism to lookup resources on network.

**Initial Context**: it provides initial context to resources.

**Session Context**: it has all the information available which is required to session bean from container.

**EjbContext**: it contain information which is required to both session and entity bean.