

KillTest

Higher Quality, Better Service!



Q&A

<http://www.killtest.com>

We offer free update service for one year.

Exam : **1Z0-895**

Title : Java Platform, Enterprise
Edition 6 Enterprise
JavaBeans Developer
Certified Expert Exam

Version : Demo

1.A developer needs to deliver a large-scale enterprise application that connects developer chooses an EJB 3.1-compliant application server, which three are true about the EJB business component tier? (Choose three.)

- A. Load-balancing is NOT a guarantee for all EJB 3.1 containers.
- B. Clustering is guaranteed to be supported by the EJB 3.1 container.
- C. Thread pooling can be optimized by the Bean Provider programmatically.
- D. Bean Providers are NOT required to write code for transaction demarcation.
- E. Support for server fail-over is guaranteed for an EJB 3.1-compliant application server.
- F. EJB 3.1 compliant components are guaranteed to work within any Java EE 6 application server

Answer: A,C,F

Explanation:

The EJB tier hosts the business logic of a J2EE application and provides system-level services to the business components problems include state maintenance, transaction management, and availability to local and remote clients.

The EJB 3.1 specification does not address "high-end" features like clustering (not B), load-balancing (A) and fail-over (not E).

F: The target platform for EJB is Java EE.

2.A developer examines a list of potential enterprise applications and selects the most appropriate technologies to use for each application.

For which two applications is EJB an appropriate solution? (Choose two.)

- A. To render a GUI for mobile clients.
- B. As a container for web-tier components including JSP.
- C. As a Web service endpoint accessed by non-Java clients.
- D. To receive and respond to HTTP Post requests directly from a web browser.
- E. As an online shopping cart which can persist across multiple sessions with a single client.

Answer: C,E

3.Which two statements are true? (Choose two.)

- A. Typically, remotely accessible objects should be coarse-grained.
- B. If a client accesses an enterprise bean locally such access must be mediated by the EJB container.
- C. A given enterprise bean's transaction information is immutable because it is deployed across various containers.
- D. If a container provides services NOT required by the EJB specification, then that container is NOT considered to be an EJB container.
- E. An enterprise bean's transaction Information can be accessed by external tools only if the information is contained in an XML deployment descriptor.

Answer: B,D

Explanation:

D: An EJB container is nothing but the program that runs on the server and implements the EJB specifications. EJB container provides special type of the environment suitable for running the enterprise components. Enterprise beans are used in distributed applications that typically contains the business logic.

4. Assume you would like to receive notification from the container as a stateless session bean transitions to and from the ready state.

Which of the following life cycle back annotations would you use? (Choose one.)

- A. @PostConstruct, @PostDestroy
- B. @PostConstruct, @PreDestroy
- C. @PreConstruct, @PostDestroy
- D. @PostConstruct, @PostDestroy, @Remove
- E. @PostConstruct, @PreDestroy, @Remove

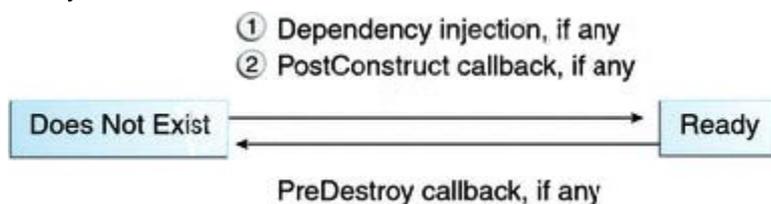
Answer: B

Explanation:

The Lifecycle of a Stateless Session Bean The EJB container typically creates and maintains a pool of stateless session beans, beginning the stateless session bean's lifecycle. The container performs any dependency injection and then invokes the method annotated @PostConstruct, if it exists. The bean is now ready to have its business methods invoked by a client.

At the end of the lifecycle, the EJB container calls the method annotated @PreDestroy, if it exists. The bean's instance is then ready for garbage collection.

Lifecycle of a Stateless Session Bean:



Note: An enterprise bean goes through various stages during its lifetime, or lifecycle. Each type of enterprise bean (stateful session, stateless session, singleton session, or message-driven) has a different lifecycle.

Reference: <http://docs.oracle.com/javaee/6/tutorial/doc/giplj.html>

5. Which API must an EJB 3.1 container make available to enterprise beans at runtime? (Choose one)

- A. The JXTA 1.1 API
- B. The MIDP 2.0 API
- C. The Java SE 6 JNDI API
- D. The Java SE 5 JDBC API

Answer: C,D

Explanation:

6. A developer wants to write a stateful session bean using the following interface as local business interface:

```
package acme;  
public interface Bar {  
    public void bar ();  
}
```

Assuming there is not an ejb-jar.xml file, which code can be inserted into Lines 4-6 below to define the bean with the ejb name of BarBean?

```
package acme;
```

```
import javax.ejb.*;
import java.io.*;
}
```

A. @Stateful

```
public class BarEJB implements Bar {
public void bar () {}
```

B. @Stateful (name = "Bar")

```
public class Barbean implements Bar {
public void bar () {}
```

C. @Stateful

```
public class BarBean implements Serializable, Bar {
public void bar () {}
```

D. @Stateful (name = "bar")

```
public class BarBean implements Serializable, Bar {
public void bar () throws java.rmi.RemoteException {}
```

Answer: C

7.A developer creates a stateful session bean that is used by many concurrent clients. The clients are written by other development team; and it is assumed that these clients might not remove the bean when ending their session. The number of concurrent sessions will be greater than the defined bean cache size. The developer must consider that the state of the session bean can be influenced by either passivation or timeout.

Which three actions should the developer take to make the bean behave correctly in passivation and timeout situations? (Choose three.)

A. Release references to resources in a @Remove annotated method.

B. Re-establish references to resources in an @Init annotated method.

C. Release references to resources in a @PreDestroy annotated method.

D. Release references to resources in a @PrePassivate annotated method.

E. Re-establish references to resources in a @PostActivate annotated method.

Answer: C,D,E

8.A stateful session bean contains a number of instance variables. The types of instance variables A and B are serializable. Instance variable B is a complex type which is populated by many business calls, and can, therefore, not be refilled by the client without starting all over. A helper instance variable C is defined as having a Serializable type, and can hold all the information which is in variable B. for example, B is of type XML-DOM tree and C of Type String.

Which two solutions, when combined, maintain the state of the session bean over a passivation and activation by the container? (Choose two.)

A. The value of helper variable C is used to create the value of Instance variable B in the beans no-arg constructor.

B. The value of helper variable C is used to create the value of instance variable B in a @postcreate annotated method.

C. The value of helper variable C is used to create the value of instance variable B in a @postActivate annotated method.

D. Instance variable A must be made null and instance variable B must be converted to a Serializable type and assigned to another instance variable in a @preDestroy annotated method.

E. Instance variable A must be defined transient. Instance variable B must be converted to a Serializable type, set to null, and assigned to the instance variable C in a @PrePassivate annotated method.

Answer: C,E

9.A developer writes a stateful session bean FooBean with one remote business interface Foo.

Foo defines an integer / setter method pair implemented as:

```
private int value;
public void setValue (int i) {value = i; }
public int getValue () {return value; }
```

A session bean ClientBean has a business method doSomething and an ejb-ref with ejb-ref-name "fooRef" that is mapped to FooBean's Foo interface.

```
@Resource private SessionContext SessionCtx;
public void doSomething () {
    Foo foo1 = (Foo) sessionCtx.lookup("fooRef");
    Foo foo2 = (Foo) sessionCtx.lookup("fooRef");
    foo1.setvalue(1);
}
```

Which statement is true after the code at line 15 completes?

A. Foo1.getValue () == 0 and foo2.getValue() == 0

B. Foo1.getValue () == 0 and foo2.getValue() == 1

C. Foo1.getValue () == 1 and foo2.getValue() == 0

D. Foo1.getValue () == 1 and foo2.getValue() == 1

Answer: D

Explanation:

Foo1 and Foo2 references the same object.

10.A developer writes a stateless session bean FooBean with one remote business interface FooRemote containing one business method foo.

Method foo takes a single parameter of application-defined type MyData.

```
public class MyData implements java.io.Serialization {
    int a;
}
```

Methods foo is implemented with the FooBean class as:

```
public void foo (MyData data) {
    data.a = 2;
}
```

Another session bean within the same application has a reference to FooRemote in variable fooRef and calls method foo with the following code:

```
MyData data = new MyData();
data.a = 1;
Fooref.foo(data);
System.out.println(data.a);
```

What is the value of data.a when control reaches Line 14 of the client?

- A. 0
- B. 1
- C. 2

Answer: B

11. Which two statements are correct about stateless session beans? (Choose two.)

- A. The bean class may declare instance variables.
- B. The lifetime of the bean instance is controlled by the client.
- C. The container may use the same bean instance to handle multiple business method invocations at the same time.
- D. The container may use the same bean instance to handle business method invocations requested by different clients, but not concurrently.

Answer: A,C

Explanation:

* A: Stateless session beans are EJB's version of the traditional transaction processing applications, which are executed using a procedure call. The procedure executes from beginning to end and then returns the result. Once the procedure is done, nothing about the data that was manipulated or the details of the request are remembered. There is no state.

These restrictions don't mean that a stateless session bean can't have instance variables and therefore some kind of internal state. There's nothing that prevents you from keeping a variable that tracks the number of times a bean has been called or that tracks data for debugging. An instance variable can even hold a reference to a live resource like a URL connection for writing debugging data, verifying credit cards, or anything else that might be useful.

C: A stateless session bean is relatively easy to develop and also very efficient. Stateless session beans require few server resources because they are neither persistent nor dedicated to one client. Because they aren't dedicated to one client, many EJB objects can use just a few instances of a stateless bean. A stateless session bean does not maintain conversational state relative to the EJB object it is servicing, so it can be swapped freely between EJB objects. As soon as a stateless instance services a method invocation, it can be swapped to another EJB object immediately. Because there is no conversational state, a stateless session bean doesn't require passivation or activation, further reducing the overhead of swapping. In short, they are lightweight and fast!

* The Lifecycle of a Stateless Session Bean Because a stateless session bean is never passivated, its lifecycle has only two stages: nonexistent and ready for the invocation of business methods. The EJB container typically creates and maintains a pool of stateless session beans, beginning the stateless session bean's lifecycle. The container performs any dependency injection and then invokes the method annotated @PostConstruct, if it exists. The bean is now ready to have its business methods invoked by a client.

At the end of the lifecycle, the EJB container calls the method annotated @PreDestroy, if it exists (not B). The bean's instance is then ready for garbage collection.

12. A developer wants to release resources within a stateless session bean class. The cleanup method should be executed by the container before an instance of the class is removed. The deployment descriptor is NOT used.

Which three statements are correct? (Choose three.)

- A. The cleanup method may declare checked exceptions.
- B. The cleanup method must have no arguments and return void.
- C. The cleanup method is executed in an unspecified transaction and security context.
- D. The developer should mark the cleanup method with the `@PreDestroy` annotation.
- E. The developer should mark the cleanup method with the `@PostDestroy` annotation.
- F. The cleanup method is executed in the transaction and security context of the last business method invocation.

Answer: B,C,D

13. A developer creates a stateless session bean. This session bean needs data from a remote system. Reading this data takes a long time. Assume that the data will NOT change during the life time of the bean and that the information to connect to the remote system is defined in JNDI.

Which statement describes how to manage the data correctly?

- A. Read the data in the bean's constructor.
- B. The data can only be read in the bean's business methods.
- C. Read the data in a method which is annotated with `@PrePassivate`.
- D. Read the data in a method which is annotated with `@PostActivate`.
- E. Read the data in a method which is annotated with `@PostConstruct`.

Answer: E

14. Suppose an EJB named `HelloWorldBean` is deployed as a standalone `ejb-jar`.

Assuming the `HelloWorldBean` is implemented as follows:

```
@LocalBean
@Stateless
public class HelloWorldBean implements HelloWorld {

    public String sayHello() {
        return generateLocalizedHello();
    }

    public String sayGoodBye() {
        return generateLocalizedGoodBye();
    }

    private String generateLocalizedHello() {
        // do some localization effort and return
    }

    private String generateLocalizedGoodBye() {
        // do some localization effort and return
    }

    // other methods
}
```

Which `HelloWorldBean` methods are accessible by another EJB within the same `ejb-jar`?

- A. All of the methods declared in `HelloWorldBean`
- B. All of the methods declared in `HelloWorld` and `HelloWorldBean`

C. All of the public methods declared in HelloWorldBean

D. All of the public methods declared in HelloWorld and all of the methods declared in HelloWorldBean

Answer: C

15. Given the following stateless session bean:

```
@Stateless
public class HelloWorldBean {

    public String sayHello() {
        return generateLocalizedHello();
    }

    public String sayGoodBye() {
        return generateLocalizedGoodBye();
    }

    private String generateLocalizedHello() {
        // do some localization effort and return
    }

    private String generateLocalizedGoodBye() {
        // do some localization effort and return
    }

    // other methods
}
```

How would you change the EJB to prevent multiple clients from simultaneously accessing the sayHello method of a single bean instance?

A. Convert sayHello into a synchronized method

B. Execute the call to generateLocalizedHello in a synchronized block

C. Convert generateLocalizehello into a synchronized method

D. Convert HelloWorldBean into a singleton bean

E. No changes are needed

Answer: A

Explanation:

* It is not possible for two invocations of synchronized methods on the same object to interleave. When one thread is executing a synchronized method for an object, all other threads that invoke synchronized methods for the same object block (suspend execution) until the first thread is done with the object.

* When a synchronized method exits, it automatically establishes a happens-before relationship with any subsequent invocation of a synchronized method for the same object. This guarantees that changes to the state of the object are visible to all threads.

Reference: The Java Tutorial, Synchronized Methods