VCE & PDF
Pass4itSure.com

1Z0-895^{Q&As}

Java EE 6 Enterprise JavaBeans Developer Certified Expert

Pass Oracle 1Z0-895 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4itsure.com/1Z0-895.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Oracle Official Exam Center

Instant Download After Purchase

100% Money Back Guarantee

365 Days Free Update

800,000+ Satisfied Customers

QUESTION 1

Suppose developer wants to create an EJB component that performs data validation every hour. Given the following Stateless session bean:

What is the minimum modification you would need to make to the bean to support notification from the TimerService once the timer expires?

A. Modify the verify external orders method to look like this: @TimedOut private void verifyExternalOrders () { / / do something }

B. Modify the verify external orders method to look like this: @EjbTimeOut private void verifyExternalOrders () { / / do something }

C. Modify the verify external orders method to look like this: @ejbTimeOut private void verifyExternalOrders () { / / do something }

D. Modify the verify external orders method to look like this:

@TimeOut

private void verifyExternalOrders () {

/ / do something

}

Correct Answer: D

Programmatic Timers

When a programmatic timer expires (goes off), the container calls the method annotated @Timeout in the bean\\'s implementation class. The @Timeout method contains the business logic that handles the timed event.

The @Timeout Method

Methods annotated @Timeout in the enterprise bean class must return void and optionally take a javax.ejb.Timer object as the only parameter. They may not throw application exceptions.

@Timeout

public void timeout(Timer timer) {

System.out.println("TimerBean: timeout occurred");

}

Reference: The Java EE 6 Tutorial, Using the Timer Service

QUESTION 2

A developer implements a CMT session bean with a method storeBoth which inserts data both a related database and an LDAP server. The relational database supports transactions while the LDAP system does NOT. Given that both updates should succeed or be rolled back, while is the best solution?

A. Implement the SessionSynchoronization interface in the session bean. In the afterCompleteion method, the LDAP inserts are rolled back if false is passed as an argument to the afterCompletion method.

B. Define the transaction attribute of the method storeBoth as REQUIRED. The container manages the transactions and will roll back modifications if something goes wrong in either database insert or LDAP insert.

C. Define the transaction attribute of the method storeBoth as REQUIRED_NEW. Carry out the database insert first. Subsequently, execute the LDAP inserts, catching LDAP exceptions. If exceptions are raised, call the SessionContext.setRollBackOnly method.

D. Define the transaction attribute of the method storeBoth as REQUIRED_NEW. Carry out the LDAP insert first. If SessionContext.getRollBackOnly returns false, execute the database inserts, catching SQL exceptions. If exceptions are raised, call the SessionContext.setRollBackOnly.

Correct Answer: C

The method should start a new transaction, so we use the REQUIRED_NEW attribute.

For the LDAP operation we can only detect LDAP exceptions. We cannot check the status of the LDAP operation through SessionContext.getRollBackOnly.

Note:

*

 CMT - Container-Managed Transactions

*

 RequiresNew Attribute

If the client is running within a transaction and invokes the enterprise bean\\'s method, the container takes the following steps:

Suspends the client\\'s transaction

Starts a new transaction

Delegates the call to the method

Resumes the client\\'s transaction after the method completes

If the client is not associated with a transaction, the container starts a new transaction before running the method.

You should use the RequiresNew attribute when you want to ensure that the method always runs within a new transaction.

Reference: The Java EE 5 Tutorial, Container-Managed Transactions

---

QUESTION 3

Which is true about caller security principal propagation for asynchronous EJB method Invocations?

A. Caller security principal propagates along with an asynchronous EJB method invocation.

B. Caller security principal does not propagate along with an asynchronous EJB method invocation.

C. Caller security principal propagates along with an asynchronous EJB method invocation only if the target bean has at least one protected method.

D. Caller security principal propagates along with an asynchronous EJB method invocation only if the calling bean has at least one protected method.

Correct Answer: D

One important caveat of asynchronous method invocation is that transactions are not propagated to the asynchronous method--a new transaction will be started for the asynchronous method. However, unlike transactions, the security principal will be propagated.

Declarative security is based only on the principal and the method being invoked, whereas programmatic security can take state into consideration.

QUESTION 4

An ejb-jar also contains three interceptor classes: AInt, BInt, CInt. Each interceptor class defines an AroundInvoke method called intercept.

The ejb-jar also contains a stateless session bean FooBean with a local business interface Foo that declares a method foo ():

10.

 @Stateless

11.

 @Intercaptors ({CInt.class, BInt.class})

12.

 public class FooBean implements Foo {

13.

14.

 public void foo () {}

15.

16.

 }

The ejb-jar contains a META-INF/ejb-jar.xml file with an section:

FooBean

com.acme.AInt

What is the interceptor order when the business methodfoo() is invoked?

A. AInt

B. AInt, CInt, BInt

C. CInt, BInt, AInt

D. AInt, BInt, CInt

Correct Answer: B

With the interceptor-order clauses Aint will be first in the order of interceptors. Within each group (default, class, method) the order of the interceptors are from left to right as defined in the @Interceptors annotation, and then the xml interceptors. In this scenario, with the @Intercaptors ({CInt.class, BInt.class}) line, the ordering continues with CInt and BInt. Note 1: By default the ordering of interceptors when invoking a method are

* External interceptors ** Default interceptors, if present ** Class interceptors, if present ** Method interceptors, if present Bean class interceptor method *

Note 2: You can override the default sort order of the external interceptors by specifiying an interceptor- binding with an interceptor-order specifying the order of the interceptors

Reference: EJB Interceptors

http://docs.jboss.org/ejb3/app-server/tutorial/interceptor/interceptor.html

QUESTION 5

Given two stateless session beans, ABean and BBean:

```
@Stateless
public class ABean {

  @EJB BBean bRef;

  public void a() {
    bRef.b();
  }

}

@Stateless
@TransactionAttribute(TransactionAttributeType.REQUIRED)
public class BBean {

  @Asynchronous
  public void b() {

}
```

A client that is not executing within a transaction acquires an EJB reference to ABean and invokes the a() method on time. How many distinct transactions are started by the container after all processing has completed?

A. 0

B. 1

C. 2

D. 3

Correct Answer: B

Note: REQUIRED If a client invokes the enterprise bean\\\'s method while the client is associated with a transaction context, the container invokes the enterprise bean\\\'s method in the client\\\'s transaction context. Reference: Enum TransactionAttributeType

To Read the Whole Q&As, please purchase the Complete Version from Our website.

Try our product !

100% Guaranteed Success
100% Money Back Guarantee
365 Days Free Update
Instant Download After Purchase
24x7 Customer Support
Average 99.9% Success Rate
More than 800,000 Satisfied Customers Worldwide
Multi-Platform capabilities - Windows, Mac, Android, iPhone, iPod, iPad, Kindle

We provide exam PDF and VCE of Cisco, Microsoft, IBM, CompTIA, Oracle and other IT Certifications.
You can view Vendor list of All Certification Exams offered:

https://www.pass4itsure.com/allproducts

Need Help

Please provide as much detail as possible so we can best assist you.
To update a previously submitted ticket: