1Z0-898

Java EE 6 Java Persistence API Developer Certified Expert

Exam Summary – Syllabus – Questions

Table of Contents

Introduction to 1Z0-898 Exam on Java EE 6 Java Persistence API Developer Certified Expert

You can use this exam guide to collect all the information about Java EE 6 Java Persistence API Developer Certified Expert (1Z0-898) certification. The Oracle 1Z0-898 certification is mainly targeted to those candidates who has some experience or exposure of Java EE 6 and want to flourish their career with Oracle Certified Expert - EE 6 Java Persistence API Developer (OCE) credential. The Java EE 6 Java Persistence API Developer Certified Expert certification exam validates your understanding of the Java EE 6 technology and sets the stage for your future progression. Your preparation plan for Oracle 1Z0-898 Certification exam should include hands-on practice or on-the-job experience performing the tasks described in following Certification Exam Topics table.

Oracle 1Z0-898 Certification Details:

Exam Name	Java EE 6 Java Persistence API Developer Certified Expert
Exam Code	1Z0-898
Exam Product Version	Java EE 6
Exam Price	USD $245 (Pricing may vary by country or by localized currency)
Duration	135
Number of Questions	64
Passing Score	61
Format	Multiple Choice
Recommended Training	Java EE 6: Develop Database Applications with JPA
Schedule Exam	Pearson VUE - Oracle
Recommended Practice	**1Z0-898 Online Practice Exam**

Oracle 1Z0-898 Exam Syllabus:

Overview of the Java Persistence API	- Describe the basics of Object Relational Mapping (ORM) - Define the key concepts of the Java Persistence API (entity, entity manager, and persistence unit)
Introducing the Auction Application	- Describe the auction application - Define the domain objects of the auction application - Describe the implementation model for the auction system

Java Persistence API Entities	- Describe the difference between objects and entities - Describe the difference between persistent fields and properties - Identify and use common Java Persistence API annotations, such as @Entity, @Id, @Table, and @Column
Understanding the Entity Manager	- Describe the relationship between an entity manager, a persistence context, and a persistence unit - Describe the difference between a container-managed entity manager and an application-managed entity manager - Describe the entity life cycle
Modeling Entity Relationships	- Examine association relationships in the data and object models - Use relationship properties to define associations - Implement one-to-one unidirectional associations - Implement one-to-one bidirectional associations - Implement many-to-one/one-to-many bidirectional associations - Implement many-to-many bidirectional associations - Implement many-to-many unidirectional associations - Examine fetch and cascade mode settings
Entity Inheritance and Object-Relational Mapping	- Examine entity inheritance - Examining object/relational inheritance hierarchy mapping strategies - Inherit from an entity class - Inherit using a mapped superclass - Inherit from a non-entity class - Examine inheritance mapping strategies - Use an embeddable class
Persisting Enums and Collections	- Persist entities that contain enums with @Enumerated - Persist entities that contain lists with @ElementCollection - Persist entities that contain maps with @ElementCollection
Introduction to Querying	- Find an Entity by its primary key - Understand basic Java Persistence API query language queries - Understand native SQL queries - Understand basic Criteria API queries
Using the Java Persistence API Query Language	- Examine the Java Persistence API query language - Create and use the SELECT statement - Create and use the UPDATE statement - Create and use the DELETE statement
Using the Java Persistence API Criteria API	- Contrast queries that use the Criteria API with queries that use the Java Persistence query language - Describe the metamodel object approach to querying - Create Criteria API queries
Using the Java Persistence API in a Container	- Use the Java Persistence API from a servlet - Use the Java Persistence API from a stateless session bean

Implementing Transactions and Locking	- Describe the transaction demarcation management - Implement container-managed transactions (CMT) - Interact programmatically with an ongoing CMT transaction - Implement bean-managed transactions (BMT) - Apply transactions to the Java Persistence API
Advanced Java Persistence API Concepts	- Specify composite primary keys - Override mappings with the @AttributeOverride and @AssociationOverride annotations - Understand entity listeners and callback methods

1Z0-898 Sample Questions:

01. Which EntityManager API will lock entity x with a pessimistic lock?
a) em.lock(x, LockModeType.WRITE)
b) em.lock(x, LockModeType.PESSIMISTIC)
c) em.lock(x, LockModeType.PESSIMISTIC_READ)
d) em.lock(x, LockModeType.OPTIMISTIC_FORCE_INCREMENT)

02. An application uses an application-managed entity manager. Which of the following is NOT true?
a) The application may specify whether the scope of the persistence context is extended.
b) The application must use EntityManagerFactory instances to create entity managers.
c) Entity manager instances must be explicitly closed.
d) The application may need to call EntityManager. joinTransaction If a JTA aware entity manager is used.

03. Which cascade option can be specified in a mapping descriptor so that it applies to all relationships in a persistent e unit?
a) cascade all
b) cascade detach
c) cascade remove
d) cascade-persist

04. If an application uses an extended persistence context, which of the following is true?
a) The persistencecontext exists until all transactions invoked by the EntityManager complete.
b) The persistence context exists until all transactions invoked by the EntityManagar complete and the EntityManager.clear () method is invoked.
c) The persistence context exists until the EntityManagerinstance is closed.
d) The persistence context exists until the EntityManagerFactory instance is closed.

05. Entity lifecycle callback methods may be defined in which three classes?
(Choose three)
a) Embedded classes
b) Entity classes
c) Abstract classes
d) Entity listener classes

e) Mapped superclasses
f) Concrete non-entity superclasses

06. A developer has created a deep entity class hierarchy with many polymorphic relationships between entitles. Which inheritance strategy, as defined by the inheritanceType enumerated type, will be most performed in this scenario?
a) Single table-per-class-hierarchy (InheritanceType.SINGLE_TABLE)
b) Joined-subclass (inheritanceType. JOINED)
c) Table-per-concrete-class (inheritanceType.TABLE_PER_CLASS)
d) Polymorphic join table (inheritanceType. POLYMORPHIC_JOIN_TABLE)

07. Which statement is correct about the Java Persistence API support for the SQL queries?
a) SQL queries are NOT allowed to use parameters.
b) The result of an SQL query is not limited to entities.
c) Only SELECT SQL queries are required to be supported.
d) SQL queries are expected to be portable across databases.

08. If a Persistence application locks entity x with a LockModeType.OPTIMISTIC_FORCE INCREMENT lock type, which statement is true?
a) The Persistence application must increment the version value prior to locking the entity.
b) This operation will result in a PersistentLockException for a non-versioned object.
c) This operation will result in a PersistentLockException if the version checks fail.
d) LockModeType.OPTIMISTIC_FORCE_INCREMENT is the synonym of the LockModeType.WRITE lock type.

09. Which two of the following statements are true of embeddable classes?
(Choose two)
a) An embeddable class must not be used to represent the state of another embeddable class.
b) Null comparison operations over embeddable classes are not supported in the Java Persistence query language.
c) An embeddable class must not contain a relationship to an entity.
d) An embeddable class can be the key of a Map relationship.

10. A developer needs to include a set of managed classes in a persistence unit. Which two solutions are correct?
(Choose two.)
a) Place the class files in the orm.xml file.
b) Place the class files in the root of the persistence unit.
c) Place the class files in any mapping file that is included on the classpath.
d) Place the class files in any jar on the classpath that is included in the persistence unit.

Answers to 1Z0-898 Exam Questions:

QUESTION: 01 Answer: c	QUESTION: 02 Answer: a	QUESTION: 03 Answer: d	QUESTION: 04 Answer: c	QUESTION: 05 Answer: b, d, e
QUESTION: 06 Answer: c	QUESTION: 07 Answer: b	QUESTION: 08 Answer: d	QUESTION: 09 Answer: b, d	QUESTION: 10 Answer: a, b

Note: If you find any typo or data entry error in these sample questions, we request you to update us by commenting on this page or write an email on feedback@oraclestudy.com