B.Hanumanthu-ISTE60

Spring Data JPA Quiz without Keys

Points:52
Time:30 min

1. What is the aim of Spring Data JPA?
a) Spring Data JPA aims to reduce the amount of boilerplate code required for common database operations.
b) Spring Data JPA aims to provide the implementation for JPA interfaces
c) Spring Data JPA is a JPA provider and implementation for JPA interfaces
d) Spring Data JPA is an ORM framework that provides an implementation for JPA interfaces

2. Which dependency is required to use Spring Data JPA in a Spring Boot application?
a) spring-boot-starter-data-jpa
b) spring-boot-starter-web
c) spring-boot-starter-test
d) spring-boot-starter-security

3. Which is the default JPA provider the Spring Data JPA internally uses?
a) EclipseLink
b) MyBatis
c) TopLink
d) Hibernate

4. Which annotation marks a class as an entity in JPA?
a) @EntityClass
b) @JPAEntity
c) @Entity
d) @TableEntity

5. Which interface do we extend to create a repository in Spring Data JPA?
a) Repository
b) JpaRepository
c) JpaRepository
d) SpringRepository

6. What is the default implementation class of the JpaRepository interface?
a) JpaRepositoryImpl class
b) SimpleJpaRepository class
c) DeafultJpaRepository class
d) SimpleCrudRepository class

7. How can you define a derived query in Spring Data JPA?
a) By annotating a method with @Query
b) By creating a method in the repository with a specific naming convention
c) By using the @DerivedQuery annotation
d) By creating an XML configuration

8. Which annotation is commonly used to mark a repository interface in Spring Data JPA?
a) @Service
b) @Component
c) @Controller
d) @Repository

9. Which annotation is used to autowire a repository into a Spring component?
a) @InjectRepository
b) @AutoInject
c) @Autowire
d) @Resource

10. Which of the following is NOT a fetch type in JPA?
a) EAGER
b) LAZY
c) IMMEDIATE
d) BOTH

11. How do you execute a native query in Spring Data JPA?
a) @Native
b) @SQL
c) @Execute
d) @Query(nativeQuery = true)

12. Which of the following is NOT a valid Cascade type in JPA?
a) PERSIST
b) REMOVE
c) MERGE
d) UPDATE

13. How can you mark a field as the primary key in a JPA entity?
a) @PrimaryKey
b) @EntityKey
c) @Id
d) @Key

14. What is the purpose of the @GeneratedValue annotation in Spring Data JPA?
a) To specify the table name for entity mapping.

b) To define the primary key column in an entity.

c) To configure the fetching strategy for related entities.

d) To automatically generate primary key values for entities.

15. How can you customize the column name for a field in a JPA entity?

a) @Column(name="custom_name")

b) @FieldName("custom_name")

c) @TableColumn("custom_name")

d) @DatabaseColumn("custom_name")

16. Which annotation is used to specify a custom query in Spring Data JPA?

a) @CustomQuery

b) @JPAQuery

c) @Query

d) @ExecuteQuery

17. Which annotation is used to create a Named JPQL query?

a) @NamedQuery

b) @NamedJPQLQuery

c) @NamedNativeQuery

d) SQLQuery

18. How can you indicate a Many-To-One relationship in JPA?

a) @ManyToOne

b) @OneToMany

c) @ManyToMany

d) @OneToOne

19. In a Spring Data JPA repository, how can you indicate a method should delete by a specific field?

a) deleteByFieldName

b) removeByFieldName

c) eraseByFieldName

d) exterminateByFieldName

20. Which of the following denotes a derived delete query in Spring Data JPA?

a) deleteByFieldName

b) findByFieldName

c) queryByFieldName

d) readByFieldName

21. How can you perform pagination in Spring Data JPA?

a) Using the @Pageable annotation

b) Using the Page and Pageable interfaces

c) Using the @Pagination annotation

d) Using the Paginator class

22. In JPA, which strategy generates a primary key assigned by the database?

a) GenerationType.AUTO

b) GenerationType.SEQUENCE

c) GenerationType.IDENTITY

d) GenerationType.TABLE

23. Which annotation in JPA allows you to embed another object as part of your entity?

a) @Embeddable

b) @Embedded

c) @Include

d) @PartOf

24. How can you control transaction management in Spring Data JPA?

a) By using the @Transactional annotation.

b) By configuring the transaction properties in the application.properties file.

c) By extending the JpaTransactionManager class.

d) By adding the @EnableTransactionManagement annotation to the configuration class

25. How can you define custom database queries in Spring Data JPA?

a) Using XML configuration files.

b) Using SQL scripts executed during application startup.

c) Using the @Query annotation and JPQL or native SQL queries.

d) Using external SQL files loaded at runtime.

26. Choose the correct Query Method:

Consider a Student entity:

```
@Entity
@Table(name = "student")
public class Student {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private int id;

    @Column(name = "first_name")
    private String firstName;

    @Transient // Field will not be saved in a database
    private String lastName;
    // getter and setter methods
```

}
Select the correct query method to find a student by firstName or LastName.

a) findByFirstNameOrLastName(String firstName, String lastName)
b) findByFirstNameAndLastName(String firstName, String lastName)
c) findByFirstNameOrLastName(Student student)
d) findByFirstNameLastName(String firstName, String lastName)