

## W03 HomeWork – Talha Arıç, [talharic@gmail.com](mailto:talharic@gmail.com)

### 1 – What is JPA ?

JPA is the Java ORM specification for permanently storing, retrieving, finding, deleting and updating data.

### 2 - What is the naming convention for finder methods in the Spring data repository interface ?

Spring data JPA has its own naming conventions for methods. Following these conventions we can build sophisticated queries. These conventions are also called as method name strategies. These strategies have defined set of keyword to use in method names. Based on the formed method name, method performs predefined operations.

### 3 - What is PagingAndSortingRepository ?

This interface provides a method `findAll(Pageable pageable)`, which is the key to implementing Pagination.

When using Pageable, we create a Pageable object with certain properties and we've to specify at least:

Page size

Current page number

Sorting

```
Sort sort = new Sort(new Sort.Order(Direction.ASC, "lastName"));
```

```
Pageable pageable = new PageRequest(0, 5, sort);
```

Passing the pageable object to the Spring data query will return the results in question (the first parameter of PageRequest is zero-based).

### 4 - Differentiate between `findById()` and `findOne()` ?

`findOne()` returns a reference to the entity with the given identifier. `findOne` internally invokes `EntityManager.getReference()` method. As per docs, this method will always return a proxy without hitting the database (lazily fetched). This method will throw `EntityNotFoundException` at the time of actual access if the requested entity does not exist in the database.

`findById()` method

This method will actually hit the database and return the real object mapping to a row in the database. It is EAGER loaded operation that returns null if no record exists in database.

### 5 - What is `@Query` used for ?

If you need to quickly create a JPA-based repository layer, Spring Data JPA is the right choice. You define your repository interface by extending one of the Spring Data JPA Repository interfaces. At

runtime, Spring Data JPA will create your repository implementations with the common CRUD methods. You can then perform CRUD operations without writing a single line of data access code.

But by saying so, Enterprise Applications developed using the Spring Framework often needs to execute complex queries against the database.

In such a scenario, you need to inform Spring Data JPA on what queries you need to execute. You do it using the `@Query` annotation.

## **6 - What is lazy loading in hibernate ?**

Lazy loading is a fetching technique used for all the entities in Hibernate. It decides whether to load a child class object while loading the parent class object. When we use association mapping in Hibernate, it is required to define the fetching technique. The main purpose of lazy loading is to fetch the needed objects from the database.

For example, we have a parent class, and that parent has a collection of child classes. Now, Hibernate can use lazy loading, which means it will load only the required classes, not all classes. It prevents a huge load since the entity is loaded only once when necessary. Lazy loading improves performance by avoiding unnecessary computation and reduce memory requirements.

Lazy loading can be used with all types of Hibernate mapping, i.e., one-to-one, one-to-many, many-to-one, and many-to-many.

## **7 – What is SQL injection attack ? Is Hibernate open to SQL injection attack ?**

SQL injection vulnerabilities arise when you construct database queries unsafely, and untrusted data gets interpreted as a part of the SQL query structure.

Injection attacks work because, for many applications, the only way to execute a given computation is to dynamically generate code that is in turn run by another system or component. If in the process of generating this code we use untrusted data without proper sanitization, we leave an open door for hackers to exploit.

## **8 - What is criteria API in hibernate ?**

Hibernate provides three different ways to retrieve data from database. We have already discussed HQL and native SQL queries. Now we will discuss our third option i.e. hibernate criteria queries. The criteria query API lets you build nested, structured query expressions in Java, providing a compile-time syntax checking that is not possible with a query language like HQL or SQL.

The Criteria API also includes query by example (QBE) functionality. This lets you supply example objects that contain the properties you would like to retrieve instead of having to step-by-step spell out the components of the query. It also includes projection and aggregation methods, including `count()`. Let's explore it's different features in detail.

## **9 - What Is Erlang? Why Is It Required For Rabbitmq ?**

Erlang is a general-purpose programming language and runtime environment for which RabbitMQ is built upon. OTP (Open Telecom Platform) is a large collection of libraries for Erlang.

RabbitMQ is an open source message-queuing software which helps in facilitating message exchange between 2 or more applications. The exchange of messages is done via queue which is defined by

administrator. An application can publish a message to the queue which can be retrieve or consumed by a different application.

## **10 – What is the JPQL ?**

JPQL is a powerful query language that allows you to define database queries based on your entity model. Its structure and syntax are very similar to SQL. But there is an important difference that I want to point out before I walk you through the different parts of a JPQL query.

JPQL uses the entity object model instead of database tables to define a query. That makes it very comfortable for us Java developers, but you have to keep in mind that the database still uses SQL. Hibernate, or any other JPA implementation, has to transform the JPQL query into SQL. It is, therefore, a good practice to activate the logging of the SQL statements during development to check the generated SQL statements.

## **11 – What are the steps to persist an entity object ?**

1. Add the hibernate dependency
2. Add the JDBC driver
3. Add JPA and database config in META-INF/persistence.xml
4. Create EntityManagerFactory
5. Mapping the Entity class
6. Use EntityManager to perform CRUD on Entity
7. Test the code

## **12 – What are the different types of entity mapping ?**

JPA entities are plain POJOs. Actually, they are Hibernate persistent entities. Their mappings are defined through JDK 5.0 annotations instead of hbm.xml files. A JPA 2 XML descriptor syntax for overriding is defined as well). Annotations can be split in two categories, the logical mapping annotations (describing the object model, the association between two entities etc.) and the physical mapping annotations (describing the physical schema, tables, columns, indexes, etc). We will mix annotations from both categories in the following code examples.

JPA annotations are in the `javax.persistence.*` package. Your favorite IDE can auto-complete annotations and their attributes for you (even without a specific "JPA" module, since JPA annotations are plain JDK 5 annotations).

A good and complete set of working examples can be found in the Hibernate Annotations test suite itself: most of the unit tests have been designed to represent a concrete example and be a source of inspiration for you. You can get the test suite sources in the distribution.

## **13 - What are the properties of an entity ?**

An entity can be place, person, object, event or a concept, which stores data in the database. The characteristics of entities are must have an attribute, and a unique key. Every entity is made up of some 'attributes' which represent that entity.

## **14 - Difference between CrudRepository and JpaRepository in Spring Data JPA?**

JpaRepository extends PagingAndSortingRepository which in turn extends CrudRepository.

Their main functions are:

CrudRepository mainly provides CRUD functions.

PagingAndSortingRepository provides methods to do pagination and sorting records.

JpaRepository provides some JPA-related methods such as flushing the persistence context and deleting records in a batch.

Because of the inheritance mentioned above, JpaRepository will have all the functions of CrudRepository and PagingAndSortingRepository.