

Repository

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

- Super
- Busca por Anotação
- JQPL
- Native Query
- Retornos

Como já mencionado na aula 1, o repository é uma bean pertencente ao spring e para utilização do JPA é necessário estender da biblioteca JpaRepository recebendo a entity e a chave como parâmetro de inicialização de classe.

Ao estender de JpaRepository, é disponibilizado alguns métodos padrões pelo super, são eles:

```
16 01 public interface JpaRepository<T, ID> extends PagingAndSortingRepository<T, ID> {
17 01     List<T> findAll();
18
19 01     List<T> findAll(Sort var1);
20
21 01     List<T> findById(Iterable<ID> var1);
22
23 01     <S extends T> List<S> saveAll(Iterable<S> var1);
24
25 01     void flush();
26
27 01     <S extends T> S saveAndFlush(S var1);
28
29 01     void deleteInBatch(Iterable<T> var1);
30
31 01     void deleteAllInBatch();
32
33 01     T getOne(ID var1);
34
35 01     <S extends T> List<S> findAll(Example<S> var1);
36
37 01     <S extends T> List<S> findAll(Example<S> var1, Sort var2);
38 }
39
```

```
@NoRepositoryBean
01 public interface CrudRepository<T, ID> extends Repository<T, ID> {
01     <S extends T> S save(S var1);
01
01     <S extends T> Iterable<S> saveAll(Iterable<S> var1);
01
01     Optional<T> findById(ID var1);
01
01     boolean existsById(ID var1);
01
01     Iterable<T> findAll();
01
01     Iterable<T> findById(Iterable<ID> var1);
01
01     long count();
01
01     void deleteById(ID var1);
01
01     void delete(T var1);
01
01     void deleteAll(Iterable<? extends T> var1);
01
01     void deleteAll();
01
01 }

```

Notação

Também podemos criar um método que retornará de acordo com a anotação, nesse caso utilizaremos o nome da variável e os operadores na descrição dos métodos por exemplo:

```
7  @Entity
8  public class Pessoa {
9      @Id
10     private Long id;
11     private String name;
12 }

9  @Repository
10 public interface PessoaRepository extends JpaRepository<Pessoa, Long> {
11
12     List<Pessoa> findByName(String name);
13     SELETOR
14 }
15
16 REFERÊNCIA
```

Java Persistence Query Language, ele é utilizado para fazer consultas em um banco de dados relacional. A sintaxe dele é muito semelhante ao SQL. Ele retorna um objeto entidade, ao invés de um campo resultado, a partir de um banco de dados, como é no SQL convencional.

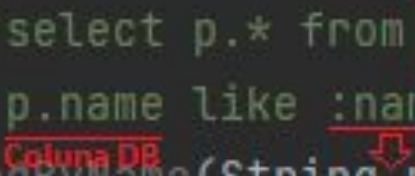
```
@Query("select p from Pessoa" + Entidade
      " where p.name like :name")
List<Pessoa> findByNome(String name);
```

Entidade
Atributo Entidade

Native Query

Native Query é uma consulta escrita como texto na linguagem de consulta do banco de dados (geralmente SQL). Se utilizado, não é garantido a estabilidade do mesmo na troca de banco (SQL Server para Oracle por exemplo).

```
@Query(value = "select p.* from pessoa" + Tabela DB  
        " where p.name like :name", nativeQuery = true)  
List<Pessoa> findByName(String name);
```



Retorno

No repository pode ser retornada uma lista do objetivo da entidade assim como valores específicos como: um único valor, colunas específicas ou uma interface com as colunas nomeadas no objeto de opcional ou não.

```
// Busca primeira pessoa cadastrada
@Query(value = "select top 1 p.* from pessoa", nativeQuery = true)
Pessoa findFirst();

// Busca o nome da primeira pessoa cadastrada
@Query(value = "select top 1 p.name from pessoa", nativeQuery = true)
String findFirstPersonName();

// Busca o nome da primeira pessoa cadastrada no objeto Optional
@Query(value = "select top 1 p.name from pessoa", nativeQuery = true)
Optional<String> findOptionalFirstPersonName();

// Busca modulando o bind para uma interface
@Query(value = "select p.name, p.email from pessoa", nativeQuery = true)
List<PessoaInterface> findPessoaToInterface();
```

```
public interface PessoaInterface {
    String getName();
    String getEmail();
}
```

AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>