JPA e Hibernate

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

- JPA
- Hibernate
- Configuração Inicial
- Anotações

JPA

Java Persistence API – JPA é uma coleção de classes e métodos voltados para armazenar persistentemente vastas quantidades de dados em um banco de dados. Com base no JPA vários FrameWorks são desenvolvidos com o objetivo de proporcionar uma interação com um banco de dados relacional, evitando com que o desenvolvedor gaste tempo com o desenvolvimento de códigos voltados para a manipulação dos dados presentes no banco de dados.

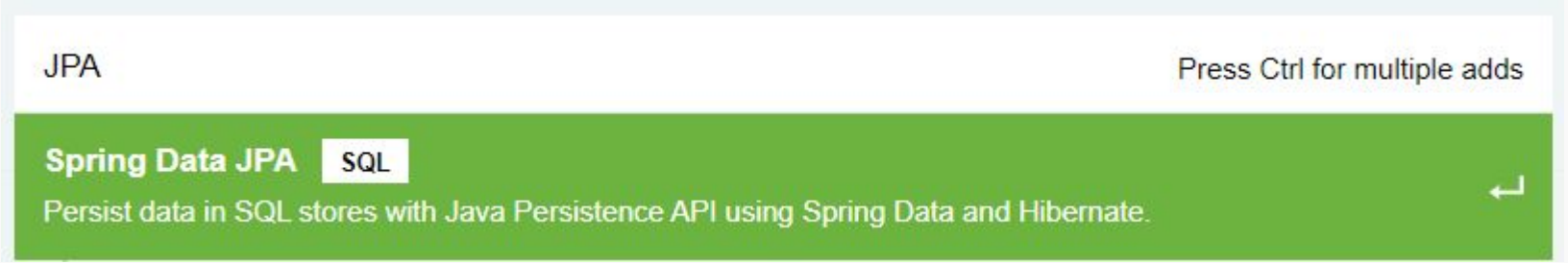
Hibernate

Hibernate é uma ferramenta para mapeamento objeto/relacional para ambientes Java. O termo mapeamento objeto/relacional (ORM) refere-se à técnica de mapeamento de uma representação de dados em um modelo de objetos para um modelo de dados relacional baseado em um esquema E/R. O Hibernate não cuida somente do mapeamento das classes Java para tabelas do banco de dados (e dos tipos de dados Java para os tipos de dados SQL), mas também provê facilidades para consulta e retornar os dados da consulta, e pode reduzir significativamente o tempo de desenvolvimento em contrapartida ao alto tempo gasto pelas operações manuais dos dados feitas com SQL e JDBC.

Configuração Inicial

Primeiramente precisamos adicionar a dependência do JPA no pom.xml. Você poderá pegar a tag de dependência pelo repositório do Maven ou iniciar com ele pelo starter. O link do repositório é: https://mvnrepository.com/artifact/org.springframework.data/spring-data-jpa

Se for pelo start.spring.io, precisará adicionar a seguinte dependência:



Configuração Inicial

Após adicionar o JPA, basta configurar o banco de dados pelo application.properties, informando a URL, usuário e senha, conforme imagem:

```
## PostgreSQL
spring.datasource.url=jdbc:postgresql://localhost:5432/postgres
spring.datasource.username=postgres
spring.datasource.password=password
```

@Entity

Para criar uma entidade, iniciamos uma classe e adicionamos a anotação @Entity. A anotação **@Entity** é utilizada para informar que uma classe também é uma entidade. A partir disso, o JPA irá efetuar a ligação entre a entidade e uma tabela de mesmo nome no banco de dados, onde os dados de objetos desse tipo poderão ser persistidos.

```java
import javax.persistence.Entity;

@Entity(name = "Pessoa")
public class PessoaEntity {

    //...

}
```

Anotação de atributos

Na entidade iremos relacionar as colunas aos atributos, para isso crie os campos e utilize (se necessário) as anotações:

@Id: Informa que o campo é uma chave primária.

@Column: Adiciona informações da coluna.

@NotNull: Informa que o campo não pode ser null.

@EmbeddedId: Caso o ID seja composto por mais de um campo e esteja em outra classe.

@JoinColumn: Adiciona referência de join para outra entidade.

@Repository

O **@Repository** tem como objetivo criar *beans* para a parte de persistência, além de capturar exceções específicas de persistência e repeti-las novamente como uma das exceções não verificadas e unificadas do Spring.

O **@Repository** é adicionado em uma Interface que por sua vez estende do JpaRepository (para efetuar a ligação com o JPA), informando a Entity e o tipo da chave.

```
public interface PessoaRepository extends JpaRepository<PessoaEntity, Long> {
    //...
}
```

Super do Repository (JPA)

Quando estendido para uso do JPA, automaticamente é fornecido os seguintes métodos:

```java
@NoRepositoryBean
public interface JpaRepository<T, ID> extends PagingAndSortingRepository<T, ID>, QueryByExampleExecutor<T> {
    List<T> findAll();

    List<T> findAll(Sort var1);

    List<T> findAllById(Iterable<ID> var1);

    <S extends T> List<S> saveAll(Iterable<S> var1);

    void flush();

    <S extends T> S saveAndFlush(S var1);

    void deleteInBatch(Iterable<T> var1);

    void deleteAllInBatch();

    T getOne(ID var1);

    <S extends T> List<S> findAll(Example<S> var1);

    <S extends T> List<S> findAll(Example<S> var1, Sort var2);
}
```

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!

SENAI

<LAB365>