



Bônus: Aprenda Fundamentos (Spring e desenvolvimento de sistemas)

Capítulo: JPA, entidades associadas, lazy loading

<https://devsuperior.com.br>

1

DISCLAIMER

Este é um **BÔNUS** concedido aos alunos do Bootcamp Spring, sem custo adicional.

O objetivo deste bônus é ensinar os seguintes fundamentos:

- Injeção de dependência
- DTO, padrão camadas
- JPA: entidades associadas, lazy loading

Vale ressaltar também que o objetivo deste bônus **NÃO É** implementar um projeto completo, mas sim abordar os fundamentos citados acima.

Bons estudos!

2

2

JPA e entidades

Sessão JPA, estados de entidades, objetos associados

3

Relembrando: estudos prévios necessários

Revisão Álgebra Relacional e SQL

Pra quê? Para relembrar as operações básicas com SQL.

<https://www.youtube.com/watch?v=GHpE5xOxXXI>

Super revisão de OO e SQL com Java e JDBC

Pra quê? Para que você compreenda na prática como é consultar os dados de um banco de dados somente com Java e JDBC, sem utilizar uma ferramenta ORM (Mapeamento Objeto-Relacional).

https://www.youtube.com/watch?v=xC_yKw3MYX4

Nivelamento ORM - JPA e Hibernate

Pra quê? Para que você tenha uma introdução teórica e prática sobre ORM com JPA, **antes de ir direto para o Spring** com o Spring Data JPA.

<https://www.youtube.com/watch?v=CAP1IPgeJkw>

4

4

JPA e gerenciamento de entidades

- A JPA faz o **gerenciamento** das entidades do sistema durante a sessão JPA.
- Uma **sessão JPA** corresponde ao contexto em que a JPA está fazendo operações com entidades durante uma conexão com o banco de dados.
 - EntityManager: objeto da JPA que encapsula uma conexão com o banco de dados e que gerencia as entidades durante uma sessão JPA.

JPA "raiz":

```
EntityManagerFactory emf = ...
EntityManager em = emf.createEntityManager();

Product prod = new ...

em.getTransaction().begin();
em.persist(prod);
em.getTransaction().commit();
```

Spring Data JPA:

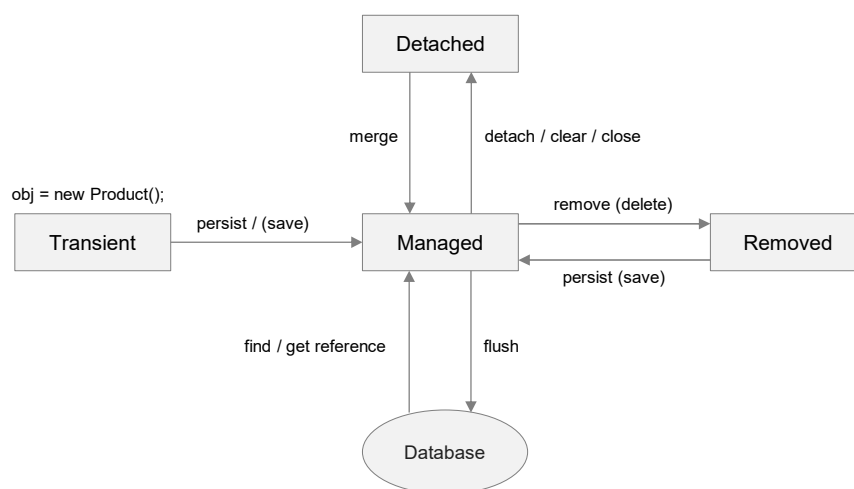
```
@Autowired
private ProductRepository repository;

@Transactional
public void meuMetodo() {
    Product prod = new ...
    repository.save(prod);
}
```

5

5

Estados de uma entidade



6

6

Salvar entidades associadas para-um

<https://github.com/devsuperior/aula-salvar-para-um>



powered by Astah

POST http://localhost:8080/people

```
{
  "name": "Nova Pessoa",
  "salary": 8000.0,
  "department": {
    "id": 1
  }
}
```

POST http://localhost:8080/people

```
{
  "name": "Nova Pessoa",
  "salary": 8000.0,
  "departmentId": 1
}
```

7

7

Salvar entidades associadas para-muitos



powered by Astah

POST http://localhost:8080/products

```
{
  "name": "Produto novo",
  "price": 1000.0,
  "categories": [
    {
      "id": 2
    },
    {
      "id": 3
    }
  ]
}
```

<https://github.com/devsuperior/aula-salvar-para-muitos>

8

8

Evitando degradação de performance

Carregamento lazy, tratativas, Transactional

9

Grande vilão da JPA

Grande vilão da JPA: idas e vindas desnecessárias ao banco de dados.

Causa comum: entidades associadas **lazy** carregando sob demanda.

10

10

Projeto exemplo

Projeto:

<https://github.com/devsuperior/aula-lazy>

Collection do Postman:

<https://www.getpostman.com/collections/d5efb333d1d308d52b7c>



powered by Astah

11

11

Carregamento lazy de entidades associadas

Carregamento padrão de entidades associadas:

- Para-um: EAGER
- Para-muitos: LAZY

Devido ao comportamento lazy, enquanto a sessão JPA estiver ativa, o acesso a um objeto associado pode provocar uma nova consulta ao banco de dados.

Formas de alterar o comportamento padrão

- Atributo fetch no relacionamento da entidade (NÃO RECOMENDADO)
- Cláusula JOIN FETCH
- Consultas customizadas (IDEAL)

12

12

Atributo fetch do relacionamento

Cuidado: a mudança desse atributo pode gerar efeitos indesejados! Evite esta abordagem!

Exemplo:

```
@ManyToOne(fetch = FetchType.LAZY)
```

13

13

Cláusula JOIN FETCH

Nota: esta cláusula não funciona para buscas paginadas do Spring.

Exemplo:

```
public interface EmployeeRepository extends JpaRepository<Employee, Long> {  
  
    @Query("SELECT obj FROM Employee obj JOIN FETCH obj.department")  
    List<Employee> findEmployeesWithDepartments();  
}
```

14

14

Cache em memória das entidades

- A JPA mantém um "cache" das entidades gerenciadas na mesma sessão JPA (mapa de identidade).
- Se você trazer entidades para a memória, a JPA não volta ao banco se você precisar novamente delas durante a mesma sessão JPA.

15

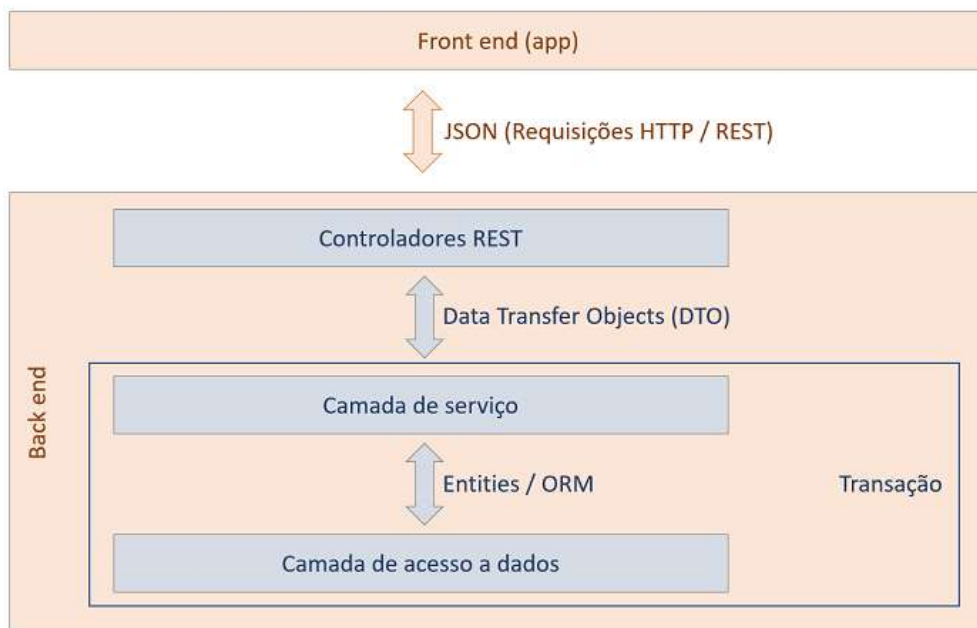
15

Transactional e open-in-view no Spring

- A annotation `@Transactional` assegura:
 - (1) resolução da transação com o banco de dados
 - (2) resolução de todas pendências "lazy" com o banco de dados
- A propriedade `spring.jpa.open-in-view=false` faz com que a sessão JPA seja encerrada antes de voltar para a camada controller (camada web)

16

16



17

Importante: problema N+1 consultas

Aulão no Youtube:

<https://www.youtube.com/watch?v=sqbqoR-IMf8>



18

18