

Spring MVC I: Criando aplicações web

9%

AULA 02

Cadastro de produtos

ATIVIDADES

01 Vídeo 121min

02 Vídeo 226min

03 Vídeo 313min

04 Cadastro de produtos

05 Sobre a utilização do @AutoWired

06 Criando o nosso controller de produtos

07 Criando o formulário de cadastro

08 Pegando os parâmetros do formulário

09 Melhorando o nosso binding

10 Adicionando o hibernate ao nosso proj...

11 Configurando o hibernate

12 Transformando a classe Produto em e...

13 Criando o ProdutoDAO

14 Usando o ProdutoDAO no Controller

15 Fazendo o spring encontrar os daos

16 Adicionando a classe JPAConfiguration ...

17 Configurando transações

AULAS

02. Cadastro de produtos

OUTROS LINKS

Fórum

Trocar Curso

Altair Lage25.1k xp

12

Transformando a classe Produto em entidade

PRÓXIMA ATIVIDADE

Agora que temos o hibernate no nosso projeto, podemos utilizá-lo para transformar as nossas classes modelos como entidades do banco de dados. Para isso precisamos anotar a classe `Produto` com a annotation `@Entity`. Além disso, o hibernate obriga que **toda entidade precisa de um *id***, ou seja, um campo que contenha um valor único para cada registro. Atualmente, a classe `Produto` não tem nenhum atributo que possa ser um *id*, portanto, crie mais um atributo para a classe `Produto` chamado `id` com o tipo `int`, e também, gere seus getters e setters.

Apesar do nome do atributo ser `id`, o hibernate ainda não entende que ele é o `id` dessa entidade. Para isso, utilizamos a annotation `@Id`, ou seja, adicione essa annotation no atributo `id`. Além disso, agora que temos esse novo atributo, precisamos populá-lo antes de persistí-lo, poderíamos fazer isso manualmente, porém, além de ser trabalhoso, é uma responsabilidade que podemos delegar ao banco de dados, ou seja, a cada vez que persistirmos um produto, o ideal é que o próprio banco já atribua um valor do `id` automaticamente. Para isso, basta apenas anotar o `id` com `@GeneratedValue`.

Por fim, precisamos informar ao hibernate como ele deve atribuir esse `id` automaticamente a partir do atributo `strategy` da annotation `@GeneratedValue`. Nesse caso, faremos com que ele seja auto incremental enviando o parâmetro `strategy = GenerationType.IDENTITY`. Cole abaixo o código da classe `Produto`.

Responda

<> INSERIR CÓDIGO

↔ EXPANDIR

ⓘ FORMATAÇÃO

TIRAR DÚVIDA

RESPONDER