

Spring MVC I: Criando aplicações web

13%

AULA 03

Cadastro de preços do produto

ATIVIDADES

01 Vídeo 123min

02 Vídeo 207min

03 Cadastro de preços do produto

04 Adicionando os preços do livro

05 Utilização do ModelAndView

06 Configuração de persistência dos preços

06 Configuração de persistência dos preços

PRÓXIMA ATIVIDADE

Por enquanto, apenas criamos a classe `Preco` e o enum `TipoPreco`. Porém, se tentarmos persistir um livro, isto é, um objeto do tipo `Produto`, não será possível, pois agora temos um atributo com referência da classe `Preco`, ou seja, por enquanto, o hibernate não sabe como lidar com objetos desse tipo. Em outras palavras, precisamos configurar o atributo `precos` da classe `Produto` para que ele seja persistido.

Inicialmente costumamos transformar as classes em entidades para que o hibernate as gerencie, entretanto, **toda entidade necessita de um id** que é uma propriedade que não faz sentido para a classe `Preco`. Para resolvermos esse tipo de problema anote a classe `Preco` com a annotation `@Embeddable`, pois ela permite que a classe `Preco` seja persistida, desde que ela seja um atributo de uma entidade, nesse caso ela é um atributo da classe `Produto` que por sinal é uma entidade.

Embora permitimos a persistência da classe `Preco`, não estamos armazenando apenas um valor dessa classe, ou seja, estamos armazenando uma lista! Para indicarmos que iremos armazenar uma lista de uma classe `@Embeddable` utilizamos a annotation `@ElementCollection`, portanto, anote o atributo `precos` da classe `Produto`.

Para o hibernate trabalhar com a tabela de preços, precisamos adicionar algumas anotações, afinal queremos que um produto tenha vários preços. Para o hibernate entender essa regra, precisamos adicionar a anotação `@ElementCollection` na nossa lista de preços e na nossa classe `Preco`, precisamos adicionar a anotação `@Embeddable`. Adicione essa anotações e cole aqui as suas classes `Produto` e `Preco`.

Responda

<> INSERIR CÓDIGO		↔ EXPANDIR	📘 FORMATAÇÃO

TIRAR DÚVIDA

RESPONDER