

Ebean com banco de dados MySQL

Transcrição

[00:00] Na aula anterior aprendemos a criar um modelo e atribuir os dados de um formulário direto nesse modelo usando um formulário de produto, `formulario.get` e temos o nosso modelo.

[00:13] Mas o modelo ainda não está sendo salvo, os dados se perdem, então precisamos salvar eles em algum lugar. Então vamos usar um banco de dados? Vamos configurar tudo, presta atenção e segue com cuidado porque essa parte é um pouquinho complicada.

[00:30] Vamos usar a ORM Ebean porque o Play já facilita o seu uso mas, para quem quiser usar o Hibernate, vamos ter uma ativ extra só para fazer a configuração do Hibernate.

[00:40] Vamos configurar o Ebean. Primeiro, precisamos adicionar o plugin do Ebean, como fizemos com o Eclipse, vamos em plugins e aqui embaixo, se você for reparar, já tem aqui uma mensagem falando “para habilitar o Ebean, descomente essa linha”, então vamos descomentar essa linha, repara que estamos usando a versão 3.0. Pode ser que esteja escrito alguma outra versão, você altera para 3.0.

[01:11] O plugin está configurado, agora precisamos habilitar ele, para isso, vamos no build sbt. Aqui no build sbt, vamos adicionar aqui dentro de “lazy val root”, vamos habilitar o plugin do PlayEbean, então temos o nosso plugin habilitado. Mas, isso não basta, nós ainda precisamos de um banco de dados, então precisamos incluir um conector de um banco MySQL nas dependências do projeto.

[01:44] Então viemos aqui em dependências e adicionamos a dependência do MySQL. Como fazemos isso? Adiciona uma vírgula aqui para indicar um novo item, abre uma nova linha e fala que queremos a dependência “mysql-connector-java”, versão, essa versão aqui que estou usando do pacote mysql.

[02:07] Então agora nós já temos o plugin do Ebean configurado, habilitado e o MySQL nas dependências do nosso projeto.

[02:16] Ainda falta uma configuração que precisamos indicar: A que banco de dados vamos nos conectar. Então vamos lá, isso acontece no `application.conf`, temos que vim aqui embaixo e habilitar algumas configurações. Primeiro, em `play.db`, você vem aqui e você remove os comentários dessas duas linhas, que indicam as configurações padrão dos bancos de dados que você vai usar no seu projeto.

[02:46] A configuração do banco de dados vai chamar “db” e o banco de dados padrão vai chamar “default”, isso é utilizado aqui embaixo, a configuração db e o banco de dados default, no caso, no seu projeto provavelmente aqui vão ter linhas diferentes, mas eu já deixei aqui pronto com as configurações que vamos usar.

[03:06] Você vem aqui embaixo em db, no bloco db e descomenta essas linhas aqui, default driver, default url, default username e default password. Com essas configurações do banco de dados MySQL e aqui definimos o nome do banco de dados, no caso vamos usar “produtos_api”.

[03:27] O username do meu banco de dados nesse computador é “root” e ele não tem senha, então é isso que vamos configurar. Falta uma coisa que precisamos fazer que é falar para o Ebean quais são os modelos que ele precisa conectar com o banco de dados, basta adicionar essa linha aqui no fim do seu arquivo de configuração da sua aplicação, “ebean.default” e passa a string aqui, “models”, dentro de colchetes.

[03:58] A parte de configuração já está pronta, então vamos criar o banco de dados. Eu vou vim aqui no MySQL, “create database produtos_api;”, agora eu vou usar a minha database produtos_api, já temos nosso banco de dados configurado, nossa database criada.

[04:22] Agora precisamos falar para o modelo que ele é um modelo de banco de dados, que ele tem que representar uma tabela de um banco de dados. Como fazemos isso? Falamos que ele é uma entidade, entity do javax persistence.

[04:39] A partir daí ele já vai ser criado no banco de dados mas, nós ainda precisamos conectar ele com o Ebean, fazemos isso estendendo a classe Model do avaje ebean.

[04:51] Ainda falta que o nosso produto tenha um id, então vamos criar aqui um id indicando que ele é gerado automaticamente e que ele é um private long que chama id.

[05:06] Olha só, agora finalmente o nosso produto está pronto. Eu vou colocar só os getters e setters aqui do id para nós conseguirmos acessar quando necessário.

[05:17] Como que geramos as tabelas no banco de dados? Então, o Play gera por nós, eu vou vim aqui e atualizar o nosso formulário e vamos ver uma mensagem, uma mensagem de que a database padrão precisa de uma evolução, essa evolução está dizendo que ele quer criar a tabela de produto.

[05:41] Pois é, a evolução é comumente chamada em outros frameworks como Rails, por exemplo, de migração, é uma coisa que talvez você já tenha ouvido falar, é basicamente uma mudança no banco de dados para adequar ele ao estado dos seus modelos. Então, aceitamos aqui que ele aplique esse script e o nosso banco de dados vai ter agora a tabela produto.

[06:05] Então, falta uma coisa só que é salvar o nosso produto no banco de dados, então viemos aqui no ProdutoController e depois de pegar o produto, usamos o método save do Ebean, daquele modelo que estendemos o produto, produto.save, só isso insere ele no banco de dados.

[06:24] Vamos ver? Viemos aqui, vamos preencher o nosso produto “livro de play”, vamos dar um código “livro-play” do tipo “livro”, vamos falar que esse é um “livro de play” e vamos dar o preço dele de 10 reais, agora cadastrando, vamos ter aquela mensagem de sucesso e podemos conferir aqui no nosso banco, select * from produto. E vemos que o nosso produto foi salvo no banco com sucesso.

[06:57] Falta uma coisinha que eu não quero deixar o meu usuário cair nessa tela feia aqui, vamos redirecionar ele de volta para o formulário de produto para ele poder criar mais um produto. Então vamos lá, basta falar aqui que vamos redirecionar, redirect, o nosso usuário para o formulário de produto.

[07:17] Como faz isso? Chamamos o arquivo de rotas. ProdutoController.formularioDeNovoProduto e assim fazemos um redirecionamento para a rota de um outro controller, ou no caso, do mesmo controller.

[07:34] Então, que que vimos nessa aula? Vimos toda configuração necessária para usar um banco de dados e também para usar o Ebean, como usar o Ebean para salvar um produto no banco com o produto.save e vimos como fazer redirecionamentos, mas o nosso formulário ainda está bem feinho.

[07:53] Nas próximas aulas, vamos ver alguns métodos de ajuda do Play Framework para criar views dinâmicas e vamos fazer isso refatorando esse formulário, vamos inclusive adicionar alguma estilização básica utilizando Bootstrap.