

Formulários e os ajudantes scala

Transcrição

[00:00] Na última aula vimos como configurar um banco de dados, criar formulários, linkar esses formulários com o nosso banco, com o nosso modelo e salvar efetivamente o produto no banco de dados, como podemos ver aqui que temos um produto dentro do nosso banco.

[00:17] Que que vamos fazer agora? O Play tem um suporte à views compiladas, como nós já tínhamos dito antes, como nós já tínhamos visto. Só que nós não estamos usando isso direito, tipo, escrevemos um código HTML puro.

[00:32] O que eu quero fazer é utilizar esses ajudantes para reduzir a nossa propensão a erros na hora de escrever os formulários e vamos ver isso agora.

[00:42] Eu quero utilizar um ajudante nas nossas views Scala, então vamos chamar o ajudante, em inglês, helper e queremos o ajudante de formulários, que é uma das maiores utilidades que temos aqui no Play.

[00:58] Então, “helper.form”, ele recebe alguns parâmetros, que tipo de parâmetros vamos receber aqui? Ele recebe uma rota e alguns outros parâmetros que nós não vamos utilizar agora.

[01:14] Rotas, nós já tínhamos visto ali no controller quando fomos redirecionar o usuário de volta para o formulário e podemos usar exatamente a mesma sintaxe aqui, que era utilizar aquele objeto de rotas que o Play já fornece para nós.

[01:29] Então é “routes.”, o controller que queremos que é o “ProdutoController” e o nome do método que é “salvaNovoProduto”. Podemos conferir aqui no nosso arquivo de rotas, salvaNovoProduto, isso vai fazer com que o nosso formulário aqui já identifique a URL e o método que estão configurados aqui nesse arquivo. No caso, /produto/novo com o método post.

[02:00] Então, substituímos esse formulário aqui por esse ajudante, eu vou apagar esse trecho aqui e adicionar essa chave para fechar o nosso ajudante. Esse ajudante vai gerar esse cabeçalho do form e vai fechar ele aqui no final também.

[02:20] Vamos então esperar ele recompilar e atualizamos aqui o nosso formulário para ver que não mudou nada, se enviarmos um novo produto, ele vai salvar no banco. Vamos criar um novo produto só para garantir, dessa vez eu vou criar aquele livro de play original, com o tipo livro, a descrição “um livro de play simples” e o preço vai ser 10 reais, enviei aqui para o cadastro, podemos conferir no banco e ele foi cadastrado.

[02:53] Nada mudou mas o nosso código está um pouquinho mais simples, um pouco menos propenso a erros porque estamos usando aqui um objeto compilado. Se nós por acaso errarmos, podemos ver aqui que existem erros de compilação.

[03:10] Mas essa não é toda a utilidade dos ajudantes que temos, o ajudante formulário permite que utilizemos os campos do nosso formulário e pegue valores já preenchidos. Então, o que precisamos fazer é passar um formulário para dentro da nossa view e utilizar os campos dele no nosso código.

[03:34] Vamos primeiro mostrar como se utiliza o código e depois corrigimos ele para se adequar às normas do Play. Para fazer o campo do título, usamos o “helper.”, isso é um input do tipo texto, então, “inputText” e passamos aqui um campo de formulário. No caso, vamos passar “formulario” e o campo de formulário pegamos desse modo, passando como parâmetro com aspas, então, “título” aqui.

[04:10] É isso. Eu vou apagar esse cara aqui e criamos um campo de texto. Só que, se viermos aqui, vamos ver que tem erro porque essa variável formulário aqui, não existe, você pode ver aqui, valor não encontrado: formulário.

[04:28] O que precisamos fazer? Precisamos passar esse formulário para a nossa view e quem é o responsável por isso é esse cara aqui, ele é o responsável por dizer que parâmetros temos dentro da nossa view. No caso, temos uma string chamada message.

[04:43] Você lembra onde vimos isso? Olha só, ProdutoController, temos isso aqui quando renderizamos a nossa view do formulário, tínhamos que passar uma string, é essa string que estamos passando ali. Então vamos alterar e vamos receber um daqueles objetos de formulário, então, “formulario” do tipo “DynamicForm”.

[05:16] Agora viemos aqui e reparamos que existe outro erro, o erro é que nós não temos um Dynamic Form sendo passado para a view, aqui temos uma string. Então precisamos gerar esse formulário, esse objeto do tipo formulário.

[05:31] Lembra como fazemos isso? Temos que pegar aquela fábrica de formulários e gerar um formulário para nós. Então, vamos fazer isso, “formulários.form”, isso gera um formulário para nós. Eu vou chamar esse cara de “formulario” e eu vou passar ele aqui para a nossa view.

[05:55] A princípio o Eclipse dá um erro porque ele não conseguiu atualizar o projeto, se nós dermos um F5 aqui, ele resolve o erro porque o Play já compilou a view e já atualizou tudo, só que o Eclipse às vezes demora um pouquinho.

[06:11] Então agora nós já temos o nosso formulário e ele já está reconhecendo que esse campo título tem que ser renderizado. Vamos dar uma atualizada aqui, olha só, ele até muda um pouquinho o estilo, ele dá uma quebrada nas linhas. Vamos substituir todos os outros então agora.

[6:31] Eu vou copiar isso aqui e substituir esses dois que são o código e o tipo. Salvando aqui e atualizando, ele já deve atualizar os dois próximos campos também.

[06:51] A descrição é uma textarea, então vamos mudar aqui, invés de inputText, vamos usar textarea com o campo descrição do nosso formulário. Atualiza aqui, ele gera uma textarea para nós com o valor de descrição, com o nome descrição.

[07:14] A mesma coisa aqui para o preço, vai ser um pouquinho diferente também, invés de um inputText, deveríamos colocar um inputNumber com preço. Deve ser isso. Tentamos atualizar e não acontece nada, aconteceu uma exceção porque o inputNumber não existe. O que precisamos fazer aqui é passar um meta parâmetro que é o tipo desse formulário. No caso, temos que passar uma apostrofe e o tipo, “type” indicando que esse tipo é um number.

[07:59] Fazendo isso, a view ainda não compila porque inputNumber não existe, temos que voltar isso aqui para inputText do tipo number. Agora ele consegue compilar e conseguimos atualizar aqui para o nosso cadastro ficar bonitinho, vê aqui que ele já reconheceu que é um tipo number e está permitindo que eu altere com os botõezinhos de mais ou menos ali.

[08:25] “Mas você tinha dito Marco que você podia preencher os valores desse formulário automaticamente” É, eu tinha dito isso mesmo. Se colocarmos qualquer valor aqui dentro desse formulário, o Play reconhece e já renderiza na nossa view com aquele valor.

[08:46] Mas como é que pomos um valor aqui dentro? O modo como valores são representados em um formulário, são mapas de string para string. Então poderíamos criar um mapa, vamos criar um mapa? “new HashMap” de string para string, guarda isso em uma variável, vamos chamar de “parametros” e vamos preencher ela.

[09:24] Como? Pomos aqui dentro dos parâmetros o tipo que é o livro. Todos os produtos que cadastramos até agora eram livros, isso quer dizer que vendemos bastante livro, então o padrão pode ser um livro.

[09:48] E como que fazemos para preencher o formulário diretamente? Mudamos essa linha aqui para baixo e falamos que o nosso formulário vai ser preenchido, vai ter um fill com esses parâmetros e então ele já vai reconhecer lá na view, já vai compilar isso e preencher o nosso formulário direto com os parâmetros. Está compilado com sucesso, atualizamos e você pode ver que o tipo já veio preenchido com o livro.

[10:20] Mas tem um jeito ainda mais legal de fazer isso, você tem um formulário dinâmico aqui mas nós já vimos que podemos fazer um formulário do tipo produto e podemos passar um produto para dentro desse formulário.

[10:35] Então vamos fazer isso, ao invés de criar um mapa de parâmetro, vamos criar um novo produto e vamos colocar dentro desse produto, o tipo livro. E aqui ao invés de criar um formulário dinâmico, vamos fazer um outro cara, vamos criar um formulário do tipo produto e vamos preencher ele com o próprio produto. Armazenamos ele aqui no formulário e passamos esse formulário para a view.

[11:19] Está vendo que deu um erro de novo? É que, na nossa view estamos recebendo um formulário dinâmico, agora temos que alterar isso para um formulário do tipo produto, isso a sintaxe que é utilizada é entre colchetes.

[11:37] Então agora se viermos aqui, está tudo compilado bonitinho, aqui ainda não atualizou, se dermos um F5 no Eclipse, resolve tudo. E agora se formos lá na view, atualizamos e ele ainda recebe um livro como parâmetro padrão.

[11:54] Por agora é só, demos uma olhada em como melhorar a nossa view e fazer ela ficar um pouco mais segura e impedindo que cometamos alguns erros, menos propensa a cometer erros. Temos poucos lugares onde usamos strings para fazer o nosso código, a maior parte do código é gerada com código compilável. Então isso é bem bacana.

[12:19] Na próxima aula, no próximo vídeo, vamos dar uma olhada em como dar uma estilizada nesse formulário para ele ficar um pouco mais bonito e ter também mensagens coloridas de validação e coisas do gênero.