

Bootstrap em uma biblioteca

Transcrição

[00:00] No vídeo anterior vimos como utilizar os ajudantes Scala do Play para simplificar um pouco a nossa view e tornar ela um pouco menos propensa a erros. Mas ela ainda está meio feinha, nosso formulário está meio cru.

[00:14] Então vamos estilizar ela, dar um estilizada nela para ela ficar um pouco mais bonitinha e podemos usar um código CSS pronto, tipo BootStrap.

[00:26] Então, na atividade que vem, eu vou disponibilizar o link de download para vocês. Mas, enquanto isso, eu vou utilizar o BootStrap que eu já baixei aqui desse mesmo link e vamos extrair ele e utilizar.

[00:42] Onde que pomos esse código do BootStrap? Naquela pasta public de recursos públicos, recursos de estilização, JavaScript, que queremos que o usuário consiga acessar sem precisar de autenticação nenhuma.

[00:58] Então eu vou extrair o BootStrap na pasta public, eu vou entrar na pasta public e eu vou mover o BootStrap para dentro dela. E eu vou dar unzip no BootStrap, agora temos uma pasta BootStrap e eu vou remover esse “zip”, bootstrap.zip.

[01:25] O BootStrap já está instalado, os estilos já estão no nosso projeto. Para importar eles, vamos ter que mexer no cabeçalho da nossa página, no head, que é onde, em geral, você importa os estilos.

[01:42] E onde fica esse cabeçalho? Lembra daquela tag main que vimos aqui, que recebe um título? Fica nela. Vamos abrir ela? “main.scala.html”, olha só, aqui estão os assets que estamos importando. Vamos trocar esses assets pelos assets do BootStrap.

[02:04] Mas primeiro deixa eu limpar um pouquinho esse código aqui e remover esses comentários também. Temos um código um pouco mais limpo.

[02:18] Agora eu dar uma modificada aqui nesse código para garantir que ele fique adequado ao BootStrap.

[02:26] Uma coisa que o BootStrap precisa é da definição de viewport, então vamos adicionar “meta name=’viewport’” com “content=’width=device-width’” e “initial-scale=1”. Isso vai definir o nosso viewport, a nossa tela. É um código que o BootStrap precisa.

[02:59] E agora temos que importar os próprios assets do BootStrap, eu vou substituir aqui esse arquivo CSS por um arquivo do BootStrap, então, “bootstrap/css/bootstrap.min.css”. E assim conseguimos importar o nosso arquivo do BootStrap.

[03:26] E porque que eu precisei só alterar esse pedacinho do código? Esse trecho aqui é uma rota, que nem a do nosso controller que indica o nosso formulário de produto, essa é uma rota que indica arquivos da pasta public. Então conseguimos acessar eles de modo um pouco mais simples, sem passar o caminho completo deles.

[03:53] Então é isso, importamos aqui o BootStrap, eu vou tirar esse JavaScript que nós não vamos usar e eu vou deixar a imagem de ícone do nosso site que é o que aparece aqui, esse iconzinho, vamos deixar ela por aí que ela é bonitinha.

[04:09] Nós atualizando aqui agora, conseguimos ver que o importe funcionou aqui, está tudo um pouco mais bonitinho porque já tem uma estilização padrão do BootStrap. Mas, está tudo muito colado nas bordas, vamos dar uma melhoria

nisso, vamos primeiro colocar um cabeçalho para garantir que esse título do nosso formulário descole do topo da página.

[04:34] Nós vamos dar uma centralizada no formulário e uma estilizada nele mas por enquanto vamos lá no cabeçalho. Para alterar o cabeçalho, podemos colocar aqui, mas eu vou querer que todas as minhas páginas tenham cabeçalho, então eu vou colocar aqui na main também.

[04:51] Importamos a main dentro de todas as nossas views e não precisa ficar se preocupando em repetir o código do cabeçalho. Eu vou usar a minha colinha aqui, a colinha do cabeçalho. O que que tem nesse cabeçalho? É uma página de navegação, você pode estudar isso um pouco mais a fundo, o Bootstrap mais a fundo, mas esse não é o foco do nosso curso.

[05:22] O que importa aqui é esse link aqui que eu criei que é uma rota para o nosso formulário de produto, então agora temos um link para um novo produto, para criar um novo produto. Vamos dar uma olhada, atualizamos aqui e, olha só, novo produto. Esse código também vai estar disponível nas atividades a seguir.

[05:44] O que mais que precisamos aqui? Podemos, olha só, eu coloquei o nosso cabeçalho no lugar errado, deixa eu colar aqui, corrigir isso e colar dentro do body. Código HTML não deve ficar fora do corpo do nosso site. Ok, devidamente indentado.

[06:05] Agora, o que eu quero aqui, eu não quero deixar esse conteúdo aqui solto, eu vou criar uma tag para ele, eu vou criar uma tag main que, sintaticamente, representa o corpo do nosso código e eu vou colar esse conteúdo aqui dentro. O que isso faz? Esteticamente, isso não faz nada mas se viermos aqui e colocar uma classe do Bootstrap nesse main, vai fazer bastante diferença.

[06:36] Eu vou colocar a classe container para garantir que isso descole das bordas, então olha só, ele ganha uma margem no topo um pouquinho mais bonitinha e descola das bordas e fica relativamente responsivo.

[06:51] Já deixando nosso formulário com uma cara bem melhor, bem mais bonita. Mas ainda tem algumas melhorias que podemos fazer. Do mesmo modo que existe o ajudante do próprio Play, podemos importar novos ajudantes, bibliotecas externas e eu encontrei uma que funciona em conjunto com o Bootstrap. Vamos colocar ela porque daí o nosso formulário vai ficar todo bonitinho.

[07:16] E como fazemos isso? Nós já vimos que com o MySQL nós precisamos importar uma biblioteca externa. A biblioteca do Bootstrap é a mesma coisa, vamos ter que vir aqui no nosso build.sbt e adicionar aqui uma dependência.

[07:34] A dependência é do “com.adrianhurt” e como que ela chama? Ela chama play bootstrap 2.11. Então, “play-bootstrap_2.11” e a versão que vamos pegar dessa biblioteca é a versão “1.0-P25”, indicando a versão do Play que estamos usando, “B3” indicando que vamos usar a versão do Bootstrap número 3.

[08:15] Vamos recarregar as nossas dependências, eu vou matar aqui o servidor, dar um reload e fazer ele carregar essa nova dependência que temos aqui. Ele já deu reload, atualizamos os arquivos do Eclipse para garantir que a nossa biblioteca chegue lá, enquanto isso, eu já vou mencionado o que que vamos fazer aqui.

[08:39] Deixa eu fechar essas coisas que não estamos usando, aqui no formulário de novo produto, temos os nossos ajudantes, para utilizar os ajudantes do Bootstrap, vamos ter que importar eles porque eles não são nativos. O helper temos nativo, os do Bootstrap não.

[09:00] Então como importamos isso? Deixa eu devolver aqui o servidor pro ar e vemos. Para importar uma biblioteca, usamos @import e o nome da biblioteca, o nome da biblioteca é “b3.vertical.fieldConstructor” esse é o pacote que estamos importando.

[09:27] Que que ele faz? Ele cria um formulário vertical utilizando os construtores de campo. Então vamos lá, vamos alterar algumas coisas aqui, primeiro, o helper não vai mais virar helper. Todos os helpers vão virar B3, que é o nome da nossa biblioteca.

[09:50] Aqui continua sendo formulário, nós passamos a nossa rota, só que aqui, invés de passar um `inputText`, vamos passar só um `text`. Ele tenta simplificar um pouquinho a nossa vida, aqui `textarea` é a mesma coisa e aqui, olha só, nós não precisamos mais do `type number`, podemos usar só `number` e ele já reconhece que isso vai ser um campo do tipo número.

[10:19] E tem mais uma coisa legal aqui, podemos criar o tipo `submit`, um botão, ele cria para nós. Nós só precisamos passar pra ele uma classe, que nem nós passamos o pseudo atributo do tipo `number` nos ajudantes do Play, passamos aqui o pseudo atributo da classe com algumas classes do Bootstrap.

[10:43] No caso, a classe `button` e a classe `button primary` para dar uma estilizada um pouco mais bonita. E o valor desse `input` vai ser o mesmo que aqui, `cadastar`. E colocamos isso entre chaves, legal, bonitinho, apagamos isso.

[11:02] Agora vamos ver lá na nossa view, damos um F5, confere aqui, compilou tudo bonitinho. Então vamos ver lá, olha só, ficou bem mais bonito já, apesar de que está um pouquinho esticado, podemos customizar isso com o pseudo elemento classe, com o pseudo atributo classe do nosso formulário, mas podemos deixar assim também.

[11:27] Mas, está faltando alguma coisa: Eu não sei mais do que que é esse campo, do que que é esse, ele perdeu os rótulos dele. Esse é um pequeno defeito dessa biblioteca que eu encontrei, que, para você ter o rótulo do título, você precisa passar o pseudo elemento, o pseudo atributo `label` e você vai escrever aqui “título”.

[11:51] A mesma coisa para todos os outros abaixo, então, aqui vai ser “código”, aqui vai ser “tipo”, “descrição” e “preço”. Vamos ver lá, conferimos, compilou bonitinho, atualiza. E ele mostra os rótulos de todos os nossos campos bonitinho, está um formulário bem mais agradável ao olho.

[12:25] Vimos nessa aula o que então? Vimos como utilizar o ajudante de formulário do Bootstrap para resolver o nosso problema de estilização, nós importamos o Bootstrap, demos uma olhada em como fazer um pedaço de código que vá aparecer em todas as nossas páginas, se quisermos, como o nosso cabeçalho aqui com o link pro novo produto. Essa é uma técnica que podemos aproveitar no futuro.

[12:53] Mas, ainda falta alguma coisa, eu ainda posso, por exemplo, tentar aqui cadastrar aquele meu livro e utilizar algumas coisas que são inadequadas. Lembra quando nós falamos que o código, ele é um identificador único do projeto? Pois é, se eu tentar cadastrar aqui esse livro, ele passa com sucesso. Ele não deveria, agora temos dois livros com o mesmo código. Isso é errado.

[13:22] É isso que vamos ver na próxima aula, vamos validar o formulário para garantir que alguns campos sejam únicos, que campos não estejam vazios e coisas desse gênero. Garantir a integridade do nosso produto, do nosso modelo.