



Criando o template

Transcrição

[00:00] Então vimos que nas páginas esse conteúdo do topo se repete e é um código mesmo repetido que se tivermos mais páginas, vai ficar mais difícil ainda de manter. Estou com a “home” e o “formulario” aberto. Se mudamos de um para o outro, nós realmente vemos que essa parte de cima é alterada, não muda, é igual.

[00:20] Então, o que vamos fazer? Se você tiver vários arquivos com esse mesmo conteúdo do topo repetido e tiver que alterar, você vai ter dificuldade. Terá que replicar em tudo. Toda alteração você terá que replicar em todos os arquivos.

[00:34] O que eu vou fazer? Em vez de ter esse código “head” aqui dentro. Nós vamos deixar o “head” vazio e eu vou jogar esse conteúdo do “head” e do “home.html” para dentro de um outro arquivo que eu vou criar aqui dentro da pasta “templates”, chamado de “base.html”. Vou copiar o arquivo.

[00:58] E aí, o que eu vou fazer? Eu vou importar esse arquivo lá na “home”. Só que para importar, eu preciso dar um nome para esse aqui. Eu vou utilizar esse atributo “fragment” do Thymeleaf para dar um nome para ele. Eu vou chamar de “head” mesmo.

[01:12] Então lá no “home” eu vou importar o “head” do arquivo “base.html”. Como fazemos isso? Nós utilizamos o “th:replace” e utilizamos o acento til nesse caso. Então tem haver com a sintaxe utilizada no Thymeleaf. Então quando você quer acessar um fragmento de uma outra página, você usa esse

com abre e fecha chaves. Aí você vai colocar qual é a página “::” e qual é o fragmento, que é “head”.

[01:50] Então, dei um “th:replace”. Eu tirei o “head” daqui da página e agora estou dando um “replace” com o conteúdo que está no “base.html” com o arquivo que nós acabamos de criar. Vou apertar a tecla “F5” no “home” e tudo permanece igual. Parece que não mudou nada.

[02:06] Vou apagar esse “replace”. Não aparece nada. Você vê que todo o estilo vem do “head”. Vou apertar a tecla “Ctrl + Z” e aí ele volta. Beleza! Nós conseguimos agora extrair da página “home” o “head” e vamos fazer a mesma coisa no “formulario”.

[02:20] Então todo esse conteúdo vai ser uma importação do “head” que está no “base.html”. Se eu clicar para abrir o “formulario”, ele está bom. Só tem um problema: na hora que eu cliquei em “Novo” e abri o formulário, olhe só, estamos com problema de “encoding”.

[02:40] Vamos só resolver isso antes de continuarmos no formulário. É esse “página”, “você deseja página aqui”. Esses dois estão com problema de “encoding”. Vamos resolver isso. Aqui no “formulario”, vou abrir “Propriedades”. Ele está definido com esse “encoding” e o melhor é utilizar o “UTF-8”.

[03:02] Vou apertar as teclas “Ctrl + X” no conteúdo dessa página, vou alterar para “UTF-8” e dar um “Apply and Close”. Ele vai pedir para eu confirmar e colar de volta aqui. Se o conteúdo estivesse aqui, ele iria converter isso para um outro caractere e iria ficar quebrado.

[03:21] Vamos ver como está agora, “formulario”. Veleza. Agora funcionou! O problema era só esse mesmo porque tínhamos um problema de “encoding” aqui. Na hora que eu criei, o arquivo “formulario” veio com esse formato e não com “UTF-8”, que era para ser o padrão.

[03:34] Você vê que o “base.html” está como “UTF-8”, que é um arquivo que acabamos de criar. Por algum motivo, o formulário foi criado com um formato diferente e deu problema agora quando utilizamos essa importação do “template”.

[03:46] Pelo menos, eu notei agora. Mas não é só isso que se repete nas páginas. Então se formos olhar aqui, não é só o “head” que tem o conteúdo de importação. Isso na verdade, visualmente não vai aparecer. Só perde o estilo.

[04:08] Nós, basicamente, quando falamos que íamos pegar o código que se repete e isolaríamos em um arquivo, estávamos falando desse código HTML aqui. O que fizemos foi isolar só os “imports” de HTML, de “style sheet”, de fonte e desse estilo que criamos para o menu, para esse da “Logo”.

[04:33] O que temos que fazer, na verdade, é isolar esse conteúdo. O da “Logo” e esse conteúdo dos “Meus Pedidos”. Então esse conteúdo de cima da “Logo”, eu vou jogar ele lá também dentro do “base” e vou colocar um nome para esse aqui também, “fragment”.

[04:53] Nós podemos criar um “fragment” com nome de “Logo”. Aí importamos para cá. Esse aqui estava aqui em cima e nós vamos importar utilizando uma “div” e o “th:replace”. Aí, de novo, tecla til que é da sintaxe quando vamos acessar um determinado fragmento, e colocamos o nome dele aqui.

[05:18] Então “base” é o arquivo, “:: Logo”. A sintaxe é essa. É só respeitar a sintaxe que vai funcionar. Vamos salvar e abrir a “home” de novo e ver se vai funcionar. Funcionou. Esse conteúdo já está vindo lá do “base.html”. E tem esse “jumbotron” também.

[05:39] Primeiro, vamos fazer a mesma coisa para o “formulario”. Vou tirar isso do formulário e jogar aqui. E agora tem o “jumbotron”, que é esse conteúdo cinza com “Meus Pedidos”. Ele também se repete no formulário, só que o título é diferente. Então aqui é “Novo Pedido” e aqui é “Meus Pedidos”.

[05:59] Quando eu for reutilizar, vou ter que passar o texto correspondendo ao “jumbotron” daquela página. Agora, como nós fazemos isso? A primeira coisa é pegarmos esse “jumbotron” e jogarmos aqui dentro. Eu vou chamar esse aqui de “titulo”, `“th:fragment = “titulo”`“.

[06:23] Só que esse “titulo” vai receber um parâmetro. Então, “valor”. Pronto! Vou apagar esse conteúdo “Meus Pedidos” e vou imprimir, na verdade, o que veio por parâmetro no “valor”. Então, pronto! Se eu quiser utilizar esse “fragment” agora, eu tenho que usar aquele “replace”, passando o nome do fragmento e tenho que passar o valor que vai ser apresentado no “titulo”.

[06:57] Então, vamos lá! Na “home”, o que eu tenho que fazer? `“Div th:replace = “~(base :: titulo())”`“. Só que eu tenho que passar o valor que eu quero que ele apresente, “Meus Pedidos”. Salvei.

[07:19] Vamos abrir a “home” de novo. Funcionou! Se eu colocar aqui, “Meus Pedidos 2”, vamos dar um “Atualizar”, aparece “Meus Pedidos 2”. Beleza, está funcionando! Então é a mesma coisa que eu vou fazer lá para o formulário.

[07:32] Eu vou substituir esse “jumbotron” que aparece “Novo Pedido” pelo “import”. Só que em vez de passar “Meus Pedidos”, eu vou passar o “Novo Pedido”. Deixe-me apagar esse e jogar o conteúdo para cá, “Novo Pedido”. Beleza! Atualizo a “home”. A “home” ficou com os “Meus Pedidos 2”. Tenho que salvar ainda. Pronto!

[07:55] Vou lá para o “formulario”. Apareceu “Novo Pedido”. Todo esse conteúdo é o que estava se repetindo entre o “formulario” e a “home”. Agora isolamos esse conteúdo e a nossa página agora fica só com o conteúdo específico que é dela. Isso facilita a alteração do código, que se repete.

[08:17] Queremos alterar, por exemplo, a versão do “Bootstrap” para uma versão mais nova, alterar o CSS do “Logo-container” ou qualquer outra coisa desse conteúdo. Nós vamos alterar e isso vai impactar todas as páginas que estiverem importando esses fragmentos.

[08:33] Então é isso. Até o próximo vídeo!