



## Modelando o status

### Transcrição

[00:00] O que nos falta implementarmos agora é o “status” do pedido. No menu temos quatro links. Um para cada “status” e um link para todos os “status”, que é o que estamos marcando aqui. Os outros “status” são “Aguardando”, “Aprovado” e “Entregue”.

[00:16] O que acontece? Quando fazemos um pedido, esse pedido automaticamente tem que ser salvo no “status”, “Aguardando”. Os compradores vão ver os pedidos que estão no “status”, “Aguardando” e vão fazer ofertas.

[00:28] Então quando uma oferta for “Aprovada” - por exemplo, você quer o “Xiaomi Redmi Note 10”, aí a pessoa faz uma oferta com um determinado valor e uma data para entregar. “Você aceita essa oferta?” Aí esse pedido entra como “status” de “Aprovado”.

[00:42] Quando o pedido é entregue, o usuário vai marcar ele como “Recebido” e ele vai para o “status” de “Entregue”. Então temos que fazer essa distinção de “status” do pedido porque visualmente tem uma classificação, que é utilizando uma coloração no “header” do “card” que envolve o pedido.

[01:01] Então a primeira alteração que vamos fazer agora para que consigamos fazer uma alteração visual aqui de acordo com o “status”, é irmos lá no pedido e adicionarmos mais um atributo aqui que representa o “status” do pedido.

[01:15] Então vou colocar “private StatusPedido status;”, vou criar essa classe que, na verdade, é um “enum”. “Create enum 'StatusPedido'” no pacote “Model” como está ali e vou adicionar o “status”. Sabemos que tem “Aguardando” é o

primeiro, “Aprovado” é quando você aprova uma determinada oferta e “Entregue” é quando você recebe o pedido. Beleza!

[01:42] Vamos salvar esse “status”. Vou salvar esse atributo como “@Enumerated”. Veja que eu não salvei o “status” do pedido sem esse “@Enumerated”. Se eu estivesse salvo aqui, ele iria reiniciar o servidor e criar a coluna com a formatação errada.

[02:01] Então antes de salvar, colocamos o “@Enumerated” como String. Se salvarmos agora, ele vai reiniciar a aplicação e vai criar essa coluna no banco de dados. Então olha na estrutura do pedido, na tabela mesmo e ele criou o “status” como “varchar”. Exatamente como queremos.

[02:19] Um pedido quando é criado, ele já é criado em um determinado “status”, que é o “Aguardando”. Então esse Xiaomi que já temos lá, nós vamos colocar como “Aguardando”. Ele já está reclamando como se eu estivesse colocando um valor errado. O “status” está como “varchar”. Pelo visto, não.

[02:58] Vamos ver qual é o problema desse aqui. Vamos colocar varchar 50. Salvamos. Volta para os dados, vamos no “status” e colocamos como “Aguardando”. Não sei o que foi que eu preenchi de errado, mas é isso. Beleza, atualizou! Não mudou nada, mas está funcionando.

[03:28] É isso o que queremos, que está conseguindo consultar no banco de dados e agora tem esse valor do “status”. O que nós vamos fazer com isso? Vamos lá na “home” e vamos categorizar os pedidos de acordo com o “status” do pedido.

[03:42] Então, como vamos fazer isso? No “header”, onde tem o nome do produto, de acordo com o exemplo que estamos usando, onde aparece o nome do produto, que é o “header” do “card” - ele tem uma coloração que vai acontecer de acordo com o “status”. Então o que eu vou fazer é mudar essa coloração de acordo com o “status”.

[04:00] Para mudar a coloração, eu já abri a página do “Bootstrap”, ele tem nos “Components Alerts” algumas configurações de cor que são as que vamos utilizar no nosso caso. Então se for o pedido que está “Aguardando”, ele vai ficar amarelo.

[04:16] E o amarelo é esse aqui, o “alert-warning”. Então se eu simplesmente salvar junto e listar de novo, ele vai aparecer como amarelo. Só que eu não quero que todos apareçam como amarelo. Eu quero que apareça de acordo com o “status”. Existe uma tag específica do “Thymeleaf”, não é do HTML, que é esse “block”.

[04:36] Ele serve apenas para você executar um comando, nós vamos executar o “switch”. E eu vou fazer um “switch” de acordo com o “status”. Então, “(pedido.status.name)”. O “name” de um “enum” é o valor do “enum” em String literal.

[04:59] Então por que eu estou passando o nome do “enum”? Porque eu vou colocar essa “div” que está dentro desse “switch”, vou colocar um “th:case”. Vou fazer o seguinte: “Apresente essa linha com o ‘warning’, caso o ‘status’ do pedido seja ‘Aguardando’”. “Aguardando” é “warning”.

[05:22] Aí tem os outros “status”. Vou só copiar e colar a linha só alterando o “status”. Então quando ele for “Aprovado”, eu quero que ele apareça com a cor verde. Essa terceira aqui, é o “success”. Então vamos colocar “alert-warning success”.

[05:44] E quando ele for “Entregue”, quero que ele fique cinza. Cinza é esse último que é o “alert-dark”. Vou colocar aqui, “alert-dark”. Então vamos salvar. Ele deu um erro. “Property or field 'status' cannot be found no pedido”. Então ele está reclamando que... Espera aí, o pedido não tem “status”! Vamos ver.

[06:11] “Status”, criar um “Getter and Setter” para esse “status” aqui e atualizar. Vamos ver o que acontece. Pronto! Ele não estava conseguindo acessar porqu

não tinha “Getter” para o “status” ainda. Só que nós ainda não temos pedidos em “status” diferentes.

[06:25] Então o que podemos fazer só para vermos é irmos lá na Amazon. Eu vou pegar um produto, vou pegar esse “Alexa”, copiar a URL do produto e colocar “Alexa” e pegar a URL da imagem. Copiar endereço da imagem e colar aqui. Vou salvar esse produto.

[06:52] Quando ele salva, ele continua abrindo essa página do formulário. Uma melhoria que podemos fazer é a seguinte: se eu conseguir criar um produto, nós vamos voltar para a página “home” e ele lista os produtos.

[07:06] Então vou atualizar o “PedidoController” para quando ele for criar um novo pedido, ele não ir para o “formulario”. E depois de salvar o pedido, ele não volta para o formulário. Ele volta para o formulário caso tenha erro. Caso ele tenha conseguido salvar o pedido, não. Então, aqui, o que fazemos?

[07:23] Existe uma String que você pode passar aqui, que é o “redirect”, que você vai pedir para o Spring fazer um “redirect” para uma outra URL “home”. Então, agora o que acontece? Depois que salvamos ele faz um “redirect” para a página “home”.

[07:40] Para experimentarmos isso, vamos cadastrar mais um outro produto. Vamos abrir o Kindle. Vou colocar a URL desse aqui. Vamos voltar para a página, para voltarmos de novo. Está dando erro de novo. O que acontece? Ele está dando erro porque o “status” do pedido não está setado no pedido novo. Beleza.

[08:09] Então, olhe só. Quando convertemos uma “requisicaoNovoPedido” para “Pedido”, agora temos que setar o “status” do pedido. Então vamos colocar “pedido.setStatus(StatusPedido.AGUARDANDO)”. Pronto! A partir desse momento não vamos mais ter problemas.

[08::30] Esse problema que aconteceu aqui é que ele está reclamando que não tem o valor e aí ele dá problema. Então para corrigirmos isso, vou só vir no banco de dados e vou só pedir para ele listar os dados de novo. Vai aparecer o “Alexa” que acabamos de criar. Ele está com “status” vazio.

[08:48] Vamos colocar “Aguardando” e atualizar ele. Beleza! Vamos atualizar de novo. Vai funcionar. Bem. Então, o que vamos fazer? Vamos criar mais um! Esse aqui do Kindle. E aí tudo vai funcionar agora. Vamos lá! “UrlProduto”, o nome do produto é “Kindle 10a”.

[09:12] Nome aqui e a “UrlImagem” do produto eu vou colocar essa mesma que dá até uma noção de tamanho para o comprador saber do que se trata e vamos cadastrar. O que esperamos? Que ele vá para a “home” e abra sem erro nenhum. Beleza! Cadê o Kindle? Está aqui. Então ele apareceu no final.

[09:35] Todos estão com “status” como “Aguardando”, mas podemos forçar outros na marra no “status”. Em vez de ficar com o “Aguardando”, eu vou colocar “Aprovado” e esse segundo eu vou colocar como “Entregue” e já salvar.

[09:56] Atualizo e vamos ver se ele mudou as cores. Beleza! Ele já está mudando as cores. “Entregue” está o Alexa, “Aprovado” está o Kindle e “Aguardando” ainda está o Xiaomi. Só que acontece o seguinte: quando é aprovado uma oferta significa que essa oferta agora tem um valor e uma data de entrega.

[10:15] Vou fazer o seguinte: só para deixarmos tudo atualizado corretamente visualmente. Esse aqui que está “Aprovado” eu vou colocar um “valorNegociado”. Esse é o Kindle. Eu nem sei qual é o valor do Kindle. Aqui está como 349 e eu vou atualizar aqui como 200.

[10:36] E a data de entrega vamos colocar para o dia 27. Vamos ver o que acontece. Beleza! Ele está mostrando agora o valor e a data de entrega. Nós podemos fazer a mesma coisa para Alexa. Eu nem sei quanto é a Alexa também. Mas vamos atualizar o banco de dados seguindo a mesma ideia.

[10:55] Vou colocar uma data de entrega para fevereiro e um valor de 450.

Pronto! Salvamos ele. Atualizamos. Pronto, já aparecem os valores. O que falta fazermos é o seguinte: esse menu ainda não está funcionando e os valores que estão aparecendo aqui estão desformatados. Isso é em reais. Tem que aparecer uma vírgula aqui, tem que aparecer o “R” e o símbolo de moeda. A data tem que aparecer com a ordem certa.

[11:25] Nós temos que aprender a fazer uma formatação de data com o “Thymeleaf”, mas isso vamos fazer no próximo vídeo. Até lá!