02

Recuperando valores com JavaScript

Transcrição

[00:00] A criação vai ser um pouco mais simples do que o que fizemos no vídeo anterior, daquele mapeamento que fizemos que envolveu todas essas classes que abrimos - que gerou, inclusive, essa tabela "oferta" no banco de dados, associada ao "pedido", que ainda não tem nada, porque não geramos nenhuma oferta.

[00:21] O que vamos fazer agora é o seguinte: voltando para a tela, ela tem valor, data da entrega e vamos adicionar o campo para comentário. Precisamos associar e receber esses dados aqui, e precisamos de um botão para o usuário poder gerar a oferta também.

[00:43] O que vamos fazer? Esse `input` que tem como `Valor` aqui, precisamos acessá-lo de alguma forma, precisamos pegar o dado dele. Nós vamos associar esse atributo `<input />` ao `pedido.` . E a que ele está associado? Ao valor negociado. Na hora em que recebemos aqui no `PedidoRest` , ele tem o atributo `ValorNegociado` . Isso é o que retornamos no JSON, então o valor negociado existe.

[01:20] O que eu vou fazer? Vou usar no Vue uma diretiva chamada `v-model` , como se estivesse fazendo uma ligação entre esse `input` , o que é digitado nesse `input` e o `pedido.valorNegociado` .

[01:36] A mesma coisa eu vou fazer para a data da entrega, só que, como estava no `Rest` ? O que ele nos retorna? `dataDaEntrega` é a data da entrega mesmo,

então é só mudar aqui no "home.html". Em vez de `valorNegociado` fica `dataDaEntrega`.

[01:54] Agora estamos *linkando* esse `input`, o valor que é preenchido pelo usuário, ao que é colocado no próprio pedido. Isso é uma coisa. A outra coisa que precisamos fazer é primeiro essa descrição aqui. O ideal é que ele fique `disabled`. Posso até atualizar a página e ver que ele fica cinza, não é visualmente a coisa mais agradável do mundo, mas tudo bem.

[02:27] Outra coisa que vamos deixar o usuário preencher, além do valor e data da entrega, é um comentário. O que eu vou fazer? Eu vou deixar os dados do produto em cima e o valor e a data da entrega eu vou puxar para baixo. Vou deixar depois do `textarea`, vou mudar um pouco o *layout* disso.

[02:48] Deixe-me recarregar a página e, beleza, tem produto, tem descrição e tudo mais. Esse conteúdo dessa linha, que tem esses dois `<div class="row">`. Eu vou só colocar aqui um *padding* (espaçamento), um *margin-top* de `3`, só para separar um pouco esse valor do que está preenchido no usuário aqui em cima.

[03:16] E o que mais nós vamos fazer é adicionar mais uma `div` aqui embaixo, que não vai ficar desabilitada, que vai estar associada a esse `<textarea class="form-control"`, vai ter o `v-model` também para `pedido.comentario`. O comentário não existe, mas ele cria e nós o acessamos depois.

[03:50] Vamos atualizar isso para vermos se funciona? Só o *text area* que ficou esquisito, o ideal é que ele tenha uma *label*. Já fizemos *label* antes, no próprio formulário nós criamos umas *labels*. Eu vou usar o mesmo esquema aqui e vou colocar no *text area* um *id*.

[04:16] No `label` nós colocamos `name` e `input`. É isso, ele criou, mas vamos deixar assim. Vou tirar esse `for="urlImagem"` e vou deixar isso de `<label>Comentário</label>`. Vamos ver como fica.

[04:43] Ainda não está a coisa mais bonita do mundo, tirando o comentário, que está estilizado, mas esses *inputs* estão bem crus. Se você parar para pensar, está bem esquisito.

[04:53] Eu vou colocar aqui, `<div class="">`. Uma `row` também, será que melhora? Vamos ver. Não. Mas se eu colocar um *margin-top* de `2`, talvez melhore. Está ruim porque ele não está alinhado, esquisito porque descrição está. Descrição não é `row`. É isso, eu e a mania do `row`!

[05:33] Fiz essa ligação, temos a tela arrumada. É isso que eu quero mostrar, esse `form-control` dá um estilo visual para o `input`, que fica mais legal. Eu vou colocar tanto no `input` do valor quanto no `input` da entrega, visualmente fica melhor. Agora temos um formulário legal.

[05:51] O que falta agora? Falta um botão. Vamos copiar esse botão aqui, `<button class="btn btn-primary" type="submit">Cadastrar</button>`, que já está estilizado. Eu vou jogá-lo aqui dentro mesmo, vou colocar, o botão será `Enviar Oferta`.

[06:09] Não precisa nem ser `type="submit"` nem nada. Não é nem formulário aqui. Vamos ver como ficou.

[06:18] Ficou colado também, não gosto disso, então *margin-top* de `2`.

[06:29] Pronto, temos valor, data da entrega, comentário, já temos o botão que podemos clicar e enviar os valores que preenchermos aqui. Mas quem vai responder esse clique no botão?

[06:43] Na documentação do Vue.js que mostrei na outra aula, existe uma opção de você colocar aqui no botão. Eu vou colocar aqui no início, o `v-on:`. O `on` representa quando alguma coisa acontece, então `on:click`. E você passa o nome de algum método que vai estar criado aqui dentro do Vue.js, aqui mesmo, dentro da sua aplicação.

[07:08] E eu vou criar aqui um `"enviarOferta(pedido)"` . Eu vou fazer a implementação mais simples, que é dando console.log, e depois nós criamos.

[07:19] Na documentação nós vimos, no outro que tem esse `mounted () {` , que é uma função que é sempre chamada. Temos esse outro objeto, `data : {` , `pedidos : [] }` , que são os dados que ficam. E existe outro objeto chamado de `methods` , onde vamos poder criar os métodos que quisermos. O método que queremos é `enviarOferta` , exatamente assim.

[07:46] Só que a implementação desse aqui não pode ser exatamente desse jeito, porque isso é uma função. Então isso tem que ter cara de função. Para isso, ter cara de função é só dizer que essa oferta é uma `function` que recebe um `(pedido)` . Eu vou colocar `console.log(pedido);` .

[08:06] Vamos só experimentar e ver se isso funciona? Então toda vez que clicarmos em enviar oferta vamos ver os dados que ele está preenchendo no pedido em questão.

[08:18] Vou carregar de novo essa tela no navegador, vou preencher o valor com "123123". A data vou colocar "10/10/2020" e o comentário vou colocar "algum comentário". Vamos apertar a tecla "F12". Eu estou no navegador, aqui na aba "Console" nós conseguimos ver tudo que vai para `console.log` .

[08:38] Ele logou aqui um objeto, esse objeto tem um monte de coisas: ele tem comentário, "algum comentário", ele criou. Tem a data de entrega, ID 6, nome do produto, ofertas, que não tem nada, só um atributo *array*. Status aguardando, URL da imagem, URL do produto e valor negociado, que é o que eu digitei aqui em cima. Esses valores eu não via antes.

[09:06] Quer ver? Vou apagar e vou clicar em "Enviar Oferta" de novo. Ele gerou outro e olhe o que tem aqui no comentário: vazio, data da entrega, vazio e valor negociado também. Então toda vez que atualizamos os campos, ele vai atualizar o próprio valor do pedido associado a esse aqui.

[09:26] E o que vamos fazer agora é a requisição para o back-end, vamos fazer validações também. Se passarmos dados inválidos nos campos "Valor" e "Data da entrega", por exemplo, isso daqui, "fdgdfg", "34545dfgdfg", conseguirmos apresentar mensagens de erro. Mas isso nós vamos fazer no próximo vídeo.