



Conclusão

Transcrição

[00:00] Para nós agora fecharmos o treinamento, vamos só fazer uma revisão no que desenvolvemos. Nós temos aqui toda uma interface gráfica construída com Bootstrap, consultando banco de dados, listando o conteúdo que veio do banco de dados e usando Spring Data JPA.

[00:16] Nós temos um formulário onde podemos preencher e ele tem todo um tratamento de mensagens de erros para apresentar para o usuário. Eu cadastrei vários pedidos diferentes para vermos, por exemplo, em estados diferentes ele funcionando direito. Cadastrei alguns pedidos.

[00:39] E no back-end nós temos uma série de tecnologias que estamos trabalhando para que possamos continuar, inclusive, o desenvolvimento das nossas funcionalidades da aplicação. Com o que já temos configurado, já conseguimos fazer bastante coisas.

[00:55] Então nós temos, começando de cima para baixo, um “Controller” principal, que é o “HomeController”. É ele quem recebe uma requisição, um “GetMapping” para “home”. Ele vai utilizando Spring Data, vai no banco de dados, pegando os pedidos e mandando para a “view” chamada de “home”. Vamos rever a “view” também.

[01:21] E temos consultas por “status”. Quando o “status” vier em um formato inválido que não existe, que vai dar erro nesse caso, aí nós tratamos essa exceção gerada nesse “ExceptionHandler”. No “PedidoController” nós temos um “GetMapping”, uma requisição para “/pedido/formulario” que só redireciona para o formulário.

[01:45] Depois que o usuário preencher o formulário, ele vai fazer uma requisição que é do tipo “post” também para “pedido”, uma requisição para um novo pedido - e é aqui que fazemos as validações dos dados para podermos apresentar os erros para o usuário. Se tudo estiver certo, nós criamos um pedido, salvamos no banco de dados e redirecionamos o usuário para “home”.

[02:09] Aqui na parte de visão, temos o “base.html”, que é o principal que tem três “fragments” diferentes que são utilizados pelas “views”. A “view home”, por exemplo, importa, inclusive, essas três - como todas vão fazer a mesma coisa. Ela importa o “head”.

[02:27] No “head”, ele pega todos os scripts que precisamos, por exemplo, do Bootstrap e do “fonts” - que aprendemos a lidar com fontes. Um estilo que desenvolvemos para ajustar algumas coisas no nosso site. A “Logo” também mostra a logomarca, aquela barra azul lá de cima. Também tem esse “fragment” que recebe um parâmetro.

[02:50] No caso do “home”, esse “fragment” chamado de “titulo” recebe um conteúdo que ele vai imprimir no topo do site. Aí utilizamos vários recursos do Thymeleaf. Por exemplo: um “th:switchcase”, um “switch case”, basicamente.

[03:11] Utilizamos também um “for it”, percorrendo a lista de pedidos que fomos buscar no banco de dados. Nós utilizamos o “th:if” também para mostrar um conteúdo ou não. Nós fizemos formatação de números no “Decimal”.

[03:31] Assim como fizemos formatação de data utilizando mais recursos do “Thymeleaf” e utilizamos esse “th:classappend” para adicionarmos um “class” ou removermos esse “class” - dependendo de algum determinado parâmetro, que no caso é o “status” do pedido. Dependendo do “status”, apresentamos uma cor diferente. Tem uma cor para cada um.

[04:03] Então é isso o que desenvolvemos! É isso que instalamos nessa aplicação utilizando o Spring Boot, Spring Data JPA, Thymeleaf.

Principalmente. são essas as tecnologias que utilizamos. Em cima delas e adicionando mais, adicionando outras - como Spring Security e Rest Controllers, além de outras tecnologias que vamos utilizar avançando no conteúdo. Vamos implementar outras funcionalidades no nosso site.

[04:37] Então é isso. Vejo vocês lá. Até a próxima!