



Formatação de número e data

Transcrição

[00:00] Agora já ajustamos o “status” e o menu. Agora precisamos ajustar esses valores que estão com a formatação errada. O número e a data de entrega. O que eu vou fazer? Deixe-me aumentar a página. Vou na documentação do Thymeleaf.

[00:20] Na aba “Docs”, nós conseguimos acessar as documentações específicas do Thymeleaf e lá embaixo temos os utilitários” Aqui, “Appendix: Expression Utility Objects”, ou seja, são utilitários que podemos usar. Eu vou utilizar, por exemplo, o utilitário de “Numbers” para poder formatar o número de acordo com o símbolo de acordo com a moeda.

[00:47] No caso, todos os valores que estamos adicionando nesse sistema estão em reais e ele não sabe fazer a conversão disso. Então vai ser sempre em reais. Eu quero que apareça aqui em reais. Então, o que eu preciso fazer? O

“Numbers” tem um utilitário. Tem vários utilitários:

“Numbers.formatDecimal”, “Numbers.formatCurrency”... É esse que queremos utilizar. Vou copiar esse exemplo e vou colocar no “valor”.

[01:14] Na hora que eu apresento o “Valor” aqui dentro, eu copio o “formatCurrency”, pego o “pedido.valorNegociado” e substituo no lugar desse “num” que vem como exemplo. Se eu apertar a tecla “F5” na minha página, vai aparecer em reais.

[01:33] Isso já é suficiente? Não! Por que não é suficiente para fazermos a formatação correta desse número? Se você olhar nas propriedades do Chron

que é o Browser que eu estou utilizando, nós conseguimos em “Avançado”, alterar o [BLOQUEIO] configurado no Browser.

[01:54] Então eu vou clicar no Inglês dos Estados Unidos e vou mover para a parte superior, aí o Browser agora vai estar fazendo requisições como se a língua do usuário fosse o Inglês. Quando eu faço isso e atualizo a página, aquela função agora vai formatar o valor em dólar. Mas esse valor será sempre em reais. Então esse “formatCurrency”, essa função não funciona para nós. Existe uma outra função aqui em cima que podemos utilizar, que é esse “formatDecimal”. Ele é legal porque diz o seguinte:

[02:29] Eu tenho um número bem grande, a cada três números eu vou separar por um ponto, mas os últimos dois eu vou separar por uma vírgula - que é exatamente o que queremos fazer. Então vou copiar esse “formatDecimal” porque ele já está como exemplo e vou substituir pelo “formatCurrency”.

[02:51] Vou pegar esse “formatCurrency”, tiro daqui, pego esse “pedido.valorNegociado” e vou substituir por esse “num” do exemplo que copiamos. Vou atualizar a página e agora está aparecendo 200, só que sem o sinal de reais.

[03:10] Então vou adicionar aqui dentro mesmo o texto “R\$” com espaço concatenando com esse valor. Atualizo e pronto! Agora está bom. Agora temos que formatar a data também. Vamos lá de novo na documentação do Thymeleaf. Tem mais um utilitário aqui que é o “Calendário”.

[03:34] E nesse “Calendars”, nesse utilitário, diferentemente do “Numbers”, o objetivo dele é formatar um determinado valor de data para uma determinada formatação. Isso é dia, mês e ano, hora e minuto. Eu não vou usar a hora e minuto, mas vou copiar esse exemplo do “Calendars” e vou substituir aqui.

[03:53] Onde eu imprimo “dataDaEntrega”, agora eu vou aplicar esse utilitário que vai formatar a data da entrega. Então vou copiar esse valor

“dataDaEntrega”, substituir no “caL” do exemplo que copiamos de lá, tirar hora e minuto e o mês só tem dois valores.

[04:10] Então é dia dois números, mês dois números e ano quatro números. Vou salvar. A ideia é que tenha essa formação. Estão vendo? Dia dois, uma barra, depois o mês, dois caracteres e outra barra e depois o ano. Vou atualizar a nossa página e ver a formatação da data como fica. Ele deu erro. Deu erro por quê? Vamos lá embaixo!

[04:36] Ele está dizendo o seguinte: o método “format” onde estamos passando “LocalDate” e uma String não pode ser encontrada, “cannot be found”. No “Calendars”, nesse utilitário “Calendars”. Por quê? Porque o “format” do “Calendars” não recebe “LocalDate”. Ele recebe “Calendar” que é uma classe de data mais antiga no Java.

[05:03] Então vamos olhar na documentação do Thymeleaf para vermos se ele tem algum que receba a nova classe de data, que é o “LocalDate”. Aqui nos utilitários temos o “Calendars” que recebe - aqui em cima ele mostra - o “Java util Calendar”.

[05:22] Tem um outro que é o “Dates”, que recebe “java.util.Date”, mas não tem nenhum que receba essa nova API de data do Java 8, que é o “LocalDate” que estamos utilizando. Só que se você voltar para a página do Thymeleaf, na aba “Download” aqui embaixo, nós temos alguns downloads que podemos fazer de bibliotecas a mais do Thymeleaf.

[05:48] Uma delas é essa “Thymeleaf-extras-java8time”. Se você clicar nela, nós vamos para o GitHub do código fonte dessa biblioteca e aqui tem alguns exemplos de como utilizarmos. Nesses exemplos ele tem um método “format” que você passa um “LocalDate”, um “[LOCALDATETYPE]” e um formato igual.

[06:10] Só que em vez de usar “Calendars”, você vai usar “temporals” e coloca aqui. A ideia é que, nesse caso, agora ele aceite o “LocalDate” - ou seja, agora

ele deveria funcionar. E funcionou! Por que ele funcionou se é uma biblioteca extra?

[06:31] Porque aqui no “pom” do “mudi” do nosso projeto, quando adicionamos o Thymeleaf, nós adicionamos esse “spring-boot-starter-thymeleaf” e nessa dependência que estamos adicionando, ele já vem com essas bibliotecas extras atualizadas. Então foi só aplicar o “temporals” e pronto, apareceu corretamente o formato da data já utilizando a biblioteca nova do [JAVA TIME LOCAL DATE].

[07:00] Então é isso! Nós já conseguimos ajustar a nossa aplicação inteira. Ela já está formatando tudo corretamente e está aparecendo todos os valores, sendo que aqui está aparecendo nulo. Já viu o “th:if” que vamos usar agora? É exatamente para não aparecer valor quando não tem valor.

[07:20] Então, só para finalizarmos aqui e corrigirmos ele, esse “thtext” só deve apresentar um valor se existir um valor no “pedido.valorNegociado”. Então para simplificarmos fazemos um “th:if = “pedido.valorNegociado != null”. Será que vai funcionar? Vamos atualizar todos. Continua nulo!

[07:51] Lembrando que estamos acessando Thymeleaf. Então vamos colocar isso corretamente. Essa expressão. Atualiza, beleza! Então agora o valor não vai aparecer quebrado. “Data da entrega:” vai ser vazio mesmo porque ele não vai conseguir formatar nada. Aparece vazio porque a data está vazia e quando tem, ele formata corretamente.

[08:16] Então é isso. Até a próxima aula!