05

# Ordenação

## Transcrição

[00:00] No último vídeo vimos como fazer paginação usando as classes do Spring page, pageable, pageRequest e passando o pageable como parâmetro para o repository, que os métodos do Spring já sabem que quando chega um pageable é para fazer a paginação via JPA e a lógica de paginação é feita automaticamente.

[00:25] Além de paginação, outra coisa comum nos projetos é que às vezes o cliente quer controlar em que ordem virão os registros. Além de paginar, quero ordenar por algum atributo específico. Como nós não definimos nada, no momento está vindo de ordem crescente pelo id, pela chave primária, conforme vem do banco de dados. Mas na aula de hoje vamos mudar isso, vamos flexibilizar, deixar que o cliente consiga controlar também qual ordem ele quer, por qual atributo do tópico ele quer ordenar os registros.

[01:00] Para fazer isso, mais um parâmetro que podemos colocar no nosso método. No caso, vou criar um parâmetro chamado ordenação, que vai ser uma string. A ideia vai ser que além de passar a página e a quantidade de registros, o cliente também vai ter que passar qual o campo que ele quer ordenar. A ideia dessa string é passar o nome do campo, que seria o atributo da nossa classe tópico. Pode ser id, data, status ou qualquer outro atributo da nossa classe tópico.

[01:42] Eu declarei só um atributo, mas preciso alterar a lógica para utilizar o novo parâmetro que está chegando no método. Como eu faço isso? Na própria classe pageable, o Spring também embutiu essa questão de ordenação. Se

dermos uma olhada no método of, vamos ver que tem várias versões. Tem um, que estamos utilizando, que só recebe a página e a quantidade de elementos, outro que recebe um sort, e o terceiro recebe um tal de Direction e um string properties, que são as propriedades que você quer ordenar. É justamente o que eu quero utilizar. Mas aí tenho que passar mais dois parâmetros, o Direction, para dizer qual a direção, se é crescente ou decrescente. Eu quero que seja crescente.

[02:47] Depois da vírgula, ele recebe o parâmetro ordenação. A princípio é só isso. Só eu passar mais um parâmetro por paginação que o Spring sabe que além de ter a paginação, tem também ordenação, pelo campo que foi passado como parâmetro.

[03:16] Vou salvar e vamos ver se não precisa mexer em mais nada. Vamos testar no Postman. Vou disparar a requisição novamente sem passar o campo ordenação. Ele já dá erro. Agora, além de passar paginação, vou aumentar a quantidade, para trazer os três registros. Vou colocar a ordenação por id, que já é o padrão. Vamos ver se funciona. Eu configurei por padrão para vir de maneira crescente.

[04:05] Vamos alterar para outro campo. Quero ordenar pela data de criação. Veio ordenado, porque a data também está ordenada. Vou alterar agora para mensagem.

[04:48] Os registros que eu deixei no banco já estão todos ordenados de maneira crescente. Só para ver se vai funcionar certinho vou mudar a ordenação para ser decrescente.

[05:18] Agora vimos que está funcionando certinho. Um parâmetro a mais na url, que chamei de ordenação, e passo o atributo que quero fazer a ordenação. No nosso código, recebemos como um parâmetro no console, e o page request passamos o campo e a ordem.

[05:40] Poderíamos também deixar a ordem flexível. Eu poderia receber mais um parâmetro para dizer qual é a direção, se é crescente ou decrescente, mas eu teria que fazer um if else. Só para não complicar, vou deixar desse jeito, mas é totalmente possível também controlar isso.

[06:03] Tem só uma questão importante aqui. Tenho que passar o nome do atributo. E se eu passar o nome do atributo errado? Se eu passar, por exemplo, o atributo data. O nome do atributo é data criação. O que acontece? Dá um erro. Se você passar o nome de um atributo que não existe, o Spring joga um exception e devolve erro 500. Ele reclama que não encontrou o campo.

[06:49] Por hoje, a aula era essa. Na próxima aula vamos ver como melhorar algumas coisas, porque podemos fazer a paginação dessa maneira, mas isso é meio que manual. Tem outro jeito mais simples, mais fácil ainda de controlar a paginação usando outros recursos do Spring, mas no próximo vídeo descobrimos como fazer isso.