



## Simplificando a paginação e ordenação

### Transcrição

[00:00] Já fizemos a parte de paginação e ordenação usando o esquema do pageable do Spring. É bem fácil, na verdade. Só uma linha de código. Você cria o objeto pageable e passa os parâmetros que estão vindo como parâmetro da url, a página, a quantidade de elementos e até a ordenação, qual o campo que você quer ordenar. Mas tem uma maneira mais fácil ainda, onde você não precisa dessa linha de código. O Spring simplificou o processo.

[00:39] Vamos ver como fazemos para utilizar. A ideia é que apagamos essa linha, e ao invés de receber os três parâmetros soltos, a página, quantidade e ordenação, já conseguimos receber todos juntos em um objeto pageable. Aquele objeto pageable conseguimos receber como parâmetro direto do nosso método lista. Não preciso receber os três campos soltos. Posso receber o pageable, e dentro dele já vem todas as informações de paginação e ordenação. A ideia é deixar o método com menos parâmetros e tirar aquela linha que estava criando o pageable na mão.

[01:22] Só que tem um problema. Para o Spring conseguir pegar os parâmetros de paginação e ordenação da requisição, ele precisa que o módulo esteja habilitado no projeto, que é o módulo que faz esse suporte de pegar as coisas da web e passar para o Spring data. Esse módulo não vem habilitado no projeto por padrão. Para habilitarmos isso, quase todas as coisas que alguns módulos que vamos utilizar do Spring Boot, que não vem habilitado por padrão, temos que ir até aquela classe fórum application, que é nossa classe main.

[02:04] Em cima da classe, só tinha o `@SpringBootApplication`, agora temos que colocar uma nova anotação, que é o `@enableSpringDataWebSupport`. Com essa anotação habilitamos esse suporte, para o Spring pegar da requisição, dos parâmetros da url os campos, as informações de paginação e ordenação, e repassar isso para o Spring data.

[02:34] E pronto, fechou. Só habilitamos na classe main e recebemos direto o  `pageable` como parâmetro do método `lista`. Vamos testar no Postman. Naquela nossa url agora precisamos passar esses parâmetros de ordenação. Porém, estávamos passando em português. Para o Spring pegar automaticamente, ele precisa que os parâmetros estejam em inglês, com determinado nome. O nome da página tem que ser `page=0&size=10&sort=id,asc`. Você separa com vírgula e coloca `asc` para crescente e `desc` para decrescente. É bem simples.

[04:18] Vou disparar a requisição. Ele me trouxe os resultados ordenados de maneira decrescente pelo id. Vou trocar para `asc`, e ele me traz os resultados de forma crescente.

[04:45] Dá até para deixar mais flexível. Imagine que quero ordenar por dois campos. Quero ordenar pelo id de maneira crescente e pela data de maneira decrescente. Você pode passar o parâmetro `sort` de novo. Ele pode aparecer múltiplas vezes. O Spring vai acumular. Ele sabe que é para ordenar por múltiplos campos e em cada campo você pode dizer se é crescente ou decrescente. Você tem total flexibilidade. Quando ele fizer a chamada para a API REST, além de pagnar e dizer quantos elementos ele quer por página, e qual a página, consigo controlar a ordem e ir por múltiplos campos, sendo que cada um pode vir de maneira crescente ou decrescente, tudo misturado.

[06:05] Mais um detalhe. E se eu não passar o campo que quero ordenar? Se eu só passar a paginação? Quero a página zero e quero dez elementos. Volta para aquele padrão. O Spring ordena pela chave primária do banco de dados. Só que você também consegue deixar isso fixo. Você consegue dizer para a Spring que se não estiver vindo parâmetro de ordenação, ele tem que ordenar por esse

campo de maneira crescente ou decrescente. Você também consegue dizer qual a ordem padrão.

[06:44] No nosso código, no parâmetro `pageable`, existe uma anotação que podemos colocar chamada `@PageableDefault`. Nesse `PageableDefault` tem alguns parâmetros. Consigo dizer que `sort = id, Direction.DESC`. Com isso, ele está dizendo que a paginação default é `id` de maneira decrescente. O default é: se não estiver vindo um parâmetro de ordenação.

[07:21] Vamos fazer o teste no Postman sem mandar os parâmetros de ordenação. Ele vem ordenado pelo `id` de maneira decrescente. Se eu passar o parâmetro `&sort=dataCriacao,asc` e disparar, veio de acordo com o parâmetro, ignorando o default. É um negócio bem interessante, bem flexível.

[08:10] Inclusive, dá para controlar também o default não só da ordem, mas também da paginação, porque e se ele não passar nenhum parâmetro? Ele vai trazer todos os registros do banco de dados, porque a paginação é opcional. Posso dizer que `page = 0, size = 10`. Se ele não passar os parâmetros de paginação, traga da primeira página apenas dez registros. Consigo deixar por padrão qual é a paginação, além da ordenação. É bem poderoso, bem simples, e o seu cliente da API consegue controlar a paginação, a ordenação, e se ele não mandar os parâmetros a gente consegue controlar qual vai ser o comportamento padrão.

[09:00] Dessa maneira, demos uma simplificada nessa parte de paginação e ordenação com Spring Boot. Por hoje, esse era o conteúdo do nosso vídeo. Espero vocês na próxima aula, onde vamos ver novos recursos para melhorar um pouco nossa API e incrementar com novas possibilidades.