



Paginação

Transcrição

[00:00] Relatórios criados, cliente feliz. Basicamente o nosso projeto está indo muito bem. Porém o pessoal da Recrutei acabou de ligar para nós informando que a consulta de funcionários está demorando muito, que os funcionários da Recrutei realizam essa consulta e vão pegar um café de tão demorado que ela está. Então eles pediram para que nós déssemos uma olhada. Vamos executar o nosso projeto e vamos ver o que está acontecendo.

[00:39] Projeto iniciado, vamos lá em funcionários e vamos visualizar aqui para ver o que está acontecendo. Retornou. Obviamente o que eu estou falando para você é um caso fictício. A consulta aqui retornou relativamente rápido. Porém você pode perceber que a consulta está retornando muitos itens e como isso é uma consulta que pode crescer exponencialmente porque não sabemos quantos funcionários têm na Recrutei, ela pode cada vez ficar maior chegando no caso de 1.000, 10.000.

[01:17] E aí cada vez que tiver mais funcionários, mais vai demorar o retorno dessa consulta. Então existem pontos na sua aplicação que essas consultas vão crescer de forma exponencial e cabe a você desenvolvedor criar formas de apresentar essas consultas mais elegantes para que não demore tanto o retorno disso para o seu usuário final. E é isso que nós vamos ver agora. Então vamos lá.

[01:49] Primeiro temos colocar aqui dentro da nossa aplicação uma forma de apresentar esse retorno dessa consulta de forma quebrada, apresentar pequenos blocos de informação e não tudo. Para isso nós temos uma forma c

fazer a consulta em programação que se chama Paginação. Com isso nós conseguimos paginar uma consulta para que possamos apresentar páginas menores de um todo. Assim as consultas ficam mais rápidas.

[02:30] Quem já fez paginação dentro de alguma linguagem de programação ou até dentro do banco de dados sabe que isso é algo muito trabalhoso, porém aqui vem uma outra grande vantagem de se utilizar o Spring Data porque ele traz aqui uma forma muito simples de fazer paginação que é basicamente o quê? Um novo Repository.

[02:53] Ele te dá um Repository que é capaz de unicamente estendendo ele utilizar paginação na sua aplicação. Então veja, estamos utilizando o CrudRepository. Vamos alterar para PagingAndSortRepository. Pronto. Já podemos fazer paginação na nossa aplicação. Veja o quão simples foi de nós adicionarmos paginação dentro da nossa aplicação utilizando o Framework do Spring Data.

[03:27] Então esses são os pequenos pontos que nos fazem levar a utilizar o Framework Spring Data. Essa simplicidade, essa velocidade que nós temos em agregar valores para dentro da nossa aplicação. Mas agora eu já posso fazer paginação, mas não significa que a nossa consulta já está paginada. Então nós temos que dentro da nossa consulta fazer a paginação. Vamos lá.

[03:54] Voltando dentro do nosso projeto vamos até a Package, canto superior esquerdo, clicando em "Service > CrudFuncionarioService". Agora aqui vamos imaginar que a paginação é como se fosse um livro, vamos imaginar um livro. Quando você vai abrir um livro você pode escolher qualquer página, não necessariamente tem de ser uma página fixa, você tem a liberdade de escolher essa página.

[04:25] Então vamos imaginar que dando essa liberdade do cliente, ele também pode escolher a página que ele quer nesse livro de funcionários que nós estamos disponibilizando para ele, essa paginação de funcionários. Então pa

isso nós vamos precisar de um Scanner para que o funcionário possa escrever no console qual a página que ele deseja visualizar. Então para isso vamos lá.

[04:48] Vamos colocar aqui um `System.out.println` e vamos fazer uma pergunta para ele, (Qual página você deseja visualizar?) Agora nós vamos precisar de um Integer e vamos chamá-lo de `Pageable` e vamos pegar do Scanner o `.nextInt` porque aqui está o número inteiro que ele vai digitar lá no Console. Agora nós temos que criar uma Request de `Pageable`.

[05:36] Então temos que criar uma Request para que retorne exatamente a página que o cliente selecionou. Vamos lá, o `Pageable` tem que ser do `DataDomain` do Framework. Então `Pageable` e vamos colocar aqui o nome da variável como `pageable` e vamos criar o nosso `PageRequest`. `PageRequest` também do `springframework.data.domain`. E aqui nós temos o `.of` que nós já podemos ver que para criar isso nós vamos precisar passar alguns valores via construtor. Vamos lá.

[06:19] O primeiro valor é a página que o cliente deseja visualizar. Aqui é a quantidade de registros que vão aparecer na página. No caso você também poderia deixar isso customizado. Porém eu vou deixar aqui limitado por quê? Porque se você deixa customizado o cliente pode colocar uma quantidade que vai demorar muito. Por exemplo, ele pode colocar 100, 1000 e aí o objetivo do `Pageable` vai por água abaixo.

[06:51] Então vou colocar 5 porque 5 é o valor que a consulta vai conseguir retornar e não vai ficar muito demorado. E por fim vamos precisar criar um Sort. Então vou colocar aqui um Sort, aqui do `DataDomain` também, `page`, 5, `Sort.unsorted`. E quando nós colocamos dentro do nosso `Repository` o `PageableAndSort`, o `findAll` consegue receber esse `Pageable` por parâmetro.

[07:25] Porque ele vai pegar esse `Pageable` e vai fazer o `findAll`, a consulta por todos respeitando a Request de `Pageable`. Então a página que o cliente quer selecionar, a quantidade de registros que ele tem que retornar e qual é a ordenação. Por fim ele não vai retornar mais uma lista, um `Iterable`, ele vai

retornar um Page também do DataDomain porque esse Page está aqui com as informações que nós vamos precisar também informar ao cliente. Por quê?

[08:00] Porque você imagina que você só retorna a consulta que voltou, só os cinco itens, o cliente não vai saber que página que ele está, o cliente não vai saber quantas páginas ele pode selecionar. Então vamos lá. Primeiro nós vamos informar para o cliente quantas possibilidades de páginas que ele tem para visualizar. Para isso eu vou colocar aqui o `System.out.println` `(funcionarios)` .

[08:37] Aqui já vai mostrar o número total de páginas que tem dentro do nosso objeto. E agora eu vou mostrar também a `System.out.println("Pagina atual")` , qual é a página atual que ele está. Então qual é a página atual, qual é a página que você está pesquisando? Para que possamos saber qual é a página que o cliente está selecionando, eu posso vir na minha Page de Funcionários, e aí eu tenho um método que se chama `GetNumber`, `System.out.println("Pagina atual" + funcionarios.get())` .

[09:19] Esse `GetNumber` vai retornar qual é a página que o cliente está visualizando naquele momento. E aqui eu vou falar para ele qual é a quantidade total de elementos que ele tem dentro dessa consulta. Para pegar o número total de elementos que tem dentro da nossa tabela, temos o `System.out.println("Total elemento " + funcionarios.getTotalElements())` .

[10:01] E agora vamos executar a nossa aplicação para ver como ficou a nossa paginação. Desculpa, como colocamos o Scanner dentro do Visualizar para que possamos pegar a página que o cliente deseja, nós vamos ter que passar também o Scanner dentro do nosso Switch. Então aqui dentro do nosso método Scanner após o cliente selecionar o código 3 nós temos que passar um Scanner. Agora sim nós podemos executar a nossa aplicação. Aplicação iniciada.

[10:42] Vamos aqui, Método 2 Funcionario, 3 para visualizar. A página que eu quero é a página 0 inicial. Então ele me retorna. Eu tenho 1 página em 3, a página que eu estou agora é 0 e o número total de elementos que eu tenho são

15. Então ele me retornou de 1 a 5. Vou visualizar de novo, mas agora eu quero a página 1.

[11:10] Veja, ele falou que eu tenho 2 de 3, a página atual que eu estou é a 1 de 15 elementos e ele me retornou os registros de 6 a 10. Vou visualizar novamente e agora eu vou para a página 3, a última página. Desculpa, a página 2 porque o *array* começa de 0. Então na página 2 eu estou na de 3 de 3, a página que eu estou é a 2 porque eu comecei da 0 e o número total de elementos são 15.

[11:43] E aí veja que ele pegou os Ids de 11 a 15. Para finalizar, vejam que voltando no nosso projeto, vindo aqui no nosso Repository, no `FuncionarioRepository` veja que estávamos utilizando a `CrudRepository` e agora nós estamos utilizando a `PagingAndSortingRepository`. Será que existem outras `Repositorys`? Isso podemos ver na próxima aula.