



## Criação da projeção

### Transcrição

[00:00] Agora que nós já estamos entregando bastante coisa para o pessoal da Recrutei e bastante coisa bem rápida graças ao Spring Data, eles solicitaram mais algumas coisas aqui para nós. Solicitaram mais um relatório. E no que esse relatório tem de apresentar para o pessoal da Recrutei? Pelo funcionário eles querem saber quais são os funcionários que recebem o maior salário dentro da companhia.

[00:29] Então eles querem que retorne um novo relatório, só que esse relatório tem que apresentar apenas o Id do funcionário, o nome do funcionário e o salário que ele tem. Com isso você pode pensar, "Vou criar uma Query `findAll` e vou apresentar para eles só os atributos que eles desejam". Mas vamos entrar aqui no DBeaver para que possamos ver um negócio.

[01:04] Se você fosse fazer uma Query `findAll`, seria algo parecido com isso `SELECT * FROM funcionários`. Vamos dar o *play*. Veja, ele retorna todos os campos de funcionários. Aqui é o Id, CPF, Data de Contratação, Nome, Salário, Cargo e aqui também não tem, mas iria retornar a correlação com as unidades de trabalho que o funcionário tem.

[01:31] Aí eu te pergunto, faz sentido nós retornarmos todas essas informações da nossa consulta se nós não vamos utilizar todos os registros? Vai deixar a nossa Query um pouco mais pesada porque vai vamos estar retornando coisas que não necessariamente vamos precisar usar. Porque se nós ainda precisássemos manipular algum atributo, ok, mas nós só queremos apresentar esses valores na tela.

[01:58] Então vamos pensar, se fizéssemos uma Query mais ou menos assim "f.id, f.nome, f.salario". Vamos executar. Aqui a nossa Query fica bem mais enxuta, só trazendo os valores necessários que precisamos apresentar no nosso console para o pessoal da Recrutei. Mas tem como eu fazer algo com o Spring Data para que eu consiga retornar apenas os atributos, os campos que eu desejo? É isso que vamos ver agora.

[02:44] Vamos lá no nosso sistema e aqui a primeira coisa que nós vamos fazer é essa Query. Nós vamos criar a Query dentro da nossa estrutura de FuncionarioRepository. Para quem não sabe onde está o FuncionarioRepository, deixa eu abrir aqui a pasta principal do projeto, na pasta "Source Java > PackageRepositorys > FuncionarioRepository". E aqui nós vamos criar a nossa Query para fazer essa consulta só com os campos que nós queremos.

[03:19] É importante ressaltar aqui que você não consegue fazer isso com Derived Querys. Tem que ser escrevendo a Query aqui com `@query`. Então vamos lá. Eu não sei o que eu vou retornar. Na verdade eu sei que é uma lista, mas eu ainda não sei qual é esse tipo de lista porque Funcionarios tem todos os atributos. Então vamos deixar um `void` e vamos colocar

```
findFuncionarioSalario .
```

[04:08] Vamos colocar aqui um `@query`, vamos colocar `value` e vamos colocar que nós vamos utilizar aqui `nativeQuery`, então: `@Query(value = "", nativeQuery = true)` Por que vamos utilizar NativeQuery? Porque eu já criei essa Query no DBeaver. É exatamente isso que eu quero apresentar no meu console. Então vou pegar exatamente essa Query que eu escrevi no DBeaver e vou colocar dentro do meu `@Query`.

[04:41] Só que agora eu preciso criar alguma coisa para que eu possa retornar ao meu serviço de relatórios uma entidade que só contenha esses atributos, exatamente esses atributos que eu vou apresentar. E como que eu consigo fazer isso com o Spring Data? O Spring Data tem uma *feature* que se chama Projeção

ou *projection* que você consegue fazer projeções de entidades que contenham apenas os atributos que você deseja apresentar. E como fazemos isso?

[05:23] Criando uma interface e padronizando os atributos dessa interface com os nomes que nós queremos retornar. Então vamos criar essa projeção. Vindo aqui em Orm vamos clicar com o botão direito, "New > Interface". Precisa ser uma interface. No campo "Name", vou colocar o nome de "FuncionarioProjecao" para que saibamos que isso é uma projeção da nossa entidade de Funcionario.

[05:57] Então FuncionarioProjecao, clicaremos em "Finish". Agora eu já tenho aqui a minha interface. E como eu faço para que o Spring Data consiga saber que esse valor Id eu quero atribuir dentro de algo aqui dentro da minha entidade. Eu tenho que criar métodos utilizando exatamente o nome da entidade que tenha o prefixo Get. Então eu quero um Get dessa entidade. Então aqui se nós viermos em Funcionario nós vamos ver que o Id é Integer, o nome é String e o salário é Double.

[06:33] Então `Integer getId` que nós queremos o valor de Id aqui dentro, `String getNome` que nós vamos querer o valor de nome aqui dentro e `Double getSalario` porque nós vamos querer o valor de salário aqui dentro. Então agora que eu já tenho essa projeção, eu vou retornar uma lista do quê? Da minha projeção. Então vindo no Repositório eu vou fazer, retorna em vez de um `void` um `List` de FuncionarioProjecao. Vamos fazer um *Import*.

[07:19] Aqui então o Spring Data já vai fazer o quê? Vai fazer essa Query nativa dentro da nossa base de dados e vai fazer uma conversão para a nossa entidade projetada. E com isso conseguimos trabalhar com um escopo bem reduzido, com uma entidade reduzida, com os atributos que nós só queremos apresentar.

[07:41] Então eu vou finalizar por aqui e na próxima aula vamos ver como que nós podemos criar o relatório e utilizar apenas a nossa FuncionarioProjecao para apresentar apenas os atributos necessários para o cliente. Então até a próxima.

