



## Usando Spring Data JPA

### Transcrição

[00:00] Vamos conhecer agora um projeto chamado, superficialmente, de Spring Data que é o que já estamos utilizando no projeto. Quer dizer, estamos adicionando ele como dependência. Então quando adicionamos a dependência do JPA para utilizarmos na comunicação com o banco de dados, nós adicionamos essa dependência no “pom.xml”, que é o “spring-boot-started-data-jpa”.

[00:29] Esse “data” se refere ao projeto Spring Data. Esse projeto é definido como um guarda-chuva, um “umbrella” de subprojetos com tecnologias para acessar repositórios de banco de dados diferentes. E aqui estamos utilizando o subprojeto chamado “spring-data-jpa” porque é o que tem integração com o JPA.

[00:52] Ele tem um nível de abstração mais alto ainda. Ele torna a configuração, a comunicação com o banco de dados, ainda bem mais simples e com muitas coisas já implementadas. Como nós vamos converter o nosso “PedidoRepository” utilizando o JPA para trabalharmos integrado com o Spring Data JPA? O Spring Data JPA é um subprojeto do Spring Data que utiliza interfaces para fazer comunicação. Interfaces mesmo, “Interface”.

[01:25] E na definição, na descrição dos métodos das interfaces é que você seguindo um padrão de nomenclatura para os nomes dos métodos, é que você explica, define para o Spring Data como aquele método tem que ser implementado - e ele pelo nome vai implementar aquele método.

[01:45] Por exemplo: recupera todos os pedidos utilizando o padrão do Spring Data de nomenclatura para um método como esse, nós utilizaríamos o “findAll”. Ou seja, quando o Spring Data vê um método chamado “findAll” que recupera uma lista de pedidos, ele sabe fazer um “select\*”.

[02:03] Ele vai implementar de acordo com o nome do método o próprio método. Como ele é uma implementação subprojeto JPA, ele sabe utilizar o “EntityManager”. Ou seja, veja que nós vamos diminuindo as nossas dependências.

[02:22] Para que esse “PedidoRepository” funcione integrado com o Spring Data, nós precisamos estender uma interface chamada de “JpaRepository” e ela precisa de dois tipos genéricos. Um é o tipo que ele está lidando, estou lidando com “Pedido”. O outro é o “ID” que é utilizado na classe “Pedido”. Se você for olhar, o tipo do “ID” é “Long”. Então você tem que definir esse aqui como tipo “Long”.

[02:54] Pronto! Só que na hora que você estende “JpaRepository”, você vai herdar algumas definições de métodos que ele já tem. Uma delas é o próprio “findAll”. Nós nem sequer precisamos do método “findAll”. Então na classe “HomeController”, nós em vez de chamarmos “recuperaTodosOsPedidos”, nós fazemos um “findAll”, que significa basicamente a mesma coisa. Principalmente se o nome do “Repository” deixar mais claro.

[03:23] Por exemplo: se fizermos um “pedidoRepository.findAll”, claramente ele vai retornar uma lista de todos os pedidos. Então pronto, é só isso! Veja que é bem mais simples. Não precisa nem daquela lista. Você vê que as dependências agora são para duas classes do Spring Data JPA. Quer dizer, é a “Repository”, a “JPA Repository” que é do Spring Data. O estereótipo que é para integrar com o Spring e o próprio “Pedido”, que é um repositório de “Pedidos”. Como ele fica simples, até as dependências ficam simples.

[04:01] Se você quiser saber mais sobre o Spring Data, entre no site do Spring “spring.io”, nos “Projects” e aqui embaixo do “Spring Framework”, tem o

“Spring Data”. Ele vai se definir como eu falei como “umbrella” de projetos que contém subprojetos específicos de um determinado banco de tecnologia de banco de dados. No nosso caso, nós estamos utilizando um subprojeto chamado de “Spring Data JPA”, mas existem vários como você pode ver.

[04:33] Os módulos principais, “Spring Data Commons”: JDBC, Spring Data JPA, LDAP, MongoDB, Redis, Cassandra... Todos esses outros e tem os da comunidade também. Couchbase, Azure Cosmos DB, DynamoDB, Elasticsearch etc. Então tem vários subprojetos dentro dessa “umbrella”, desse guarda-chuva de projetos, que é o Spring Data.

[05:04] Então é isso, galera! Ficou bem mais simples. Então vamos ver se isso funciona mesmo. Vamos subir a aplicação. Beleza, não deu erro! Ela subiu direito. O que eu vou tentar fazer agora é aquela consulta.

[05:20] Beleza, ele conseguiu ir no banco de dados! Só que agora utilizando o Spring Data, ou seja, nós não vamos utilizar mais nenhuma anotação relativa especificamente ao JPA. Nós vamos lidar agora só na camada do Spring Data quando falarmos de repositório, de comunicação com o banco de dados.

[05:40] Então, até o próximo vídeo!