

Logo da Orange Talents

# Recebendo dados via DELETE

## Cenário:

Utilizando os conhecimentos em Micronaut adquiridos até este momento, imagine que precisamos implementar uma API REST para excluir um autor do banco de dados relacional. Como restrições para API REST, temos:

* Um autor é identificado de forma única por seu ID no banco de dados;
* Em caso de sucesso devemos retornar o status HTTP 200;
* Em caso de autor não encontrado devemos também retornar um status HTTP 200;

Felizmente a entidade da JPA que representa um autor já existe no projeto e está mapeada corretamente para uma tabela no banco de dados assim como seu repository do Micronaut, ou seja, não precisamos escrevê-las.

Portanto, temos duas perguntas para você:

1. Como você implementaria uma API REST com Micronaut para excluir os dados do autor?
2. Você acha que faz sentido retornar um status HTTP 200 quando o autor não existe no banco de dados? Por quê?

## O que seria bom ver nessa resposta?

* **Peso 5**: Criar um controller com um método para receber o autor respeitando as restrições da atividade e excluir a entidade do banco de dados;
* **Peso 2**: Anotar o método do controller com @Delete do Micronaut com uma URI usando o ID do autor para indicar o resource especifico;
* **Peso 3**: Explicar o motivo de não fazer diferença para esse endpoint. Melhor ainda se referenciar a prática de idempotência usado em APIs REST ou mesmo idempotência para o verbo HTTP DELETE;

## Resposta do Especialista:

* Crio uma classe de controller com um método para excluir o autor submetido na requisição. Anoto a classe com @Controller para indicar que se trata de um controller do Micronaut, e anoto o método com @Delete("/api/autores/{id}") para indicar que se trata de um endpoint para tratar uma requisição com verbo HTTP DELETE, afinal geralmente utilizamos DELETE para indicar exclusão de recursos numa API REST;
* Injeto o repository de autor no controller e o utilizo dentro do método do controller para verificar se o autor já existe no sistema pelo id extraído da URI do endpoint. Caso o autor não exista no banco de dados eu simplemente retorno um status HTTP 200 indicando que o autor foi excluído com sucesso;
* Faço mais um teste no POSTman passando um id inexistente no banco de dados para ter certeza que um status 200 seja retornado;
* Caso o autor exista no sistema, uso o repository para excluir a entidade do banco de dados e retorno um status HTTP 200 indicando que a operação ocorreu com sucesso;
* Por fim, eu testo o endpoint usando o POSTman ou Insomnia para ter certeza que tudo ocorreu como esperado;
* Sobre retornar o status HTTP 200 quando o autor não existir no sistema, não só faz sentido como é uma prática comum pois a entidade indicada na requisição pode ter sido excluída por outra requisição, ou seja, tentar excluir uma entidade que não existe **não altera o resultado final esperado pelo cliente da API REST: a entidade está excluída**. Tanto é, que denominamos essa prática de **idempotência**, que, de acordo com a especificação HTTP, é algo padrão e esperado em requisições que usam o verbo DELETE;