

Logo da Orange Talents

# Recebendo dados via PUT

## Cenário:

Utilizando os conhecimentos em Micronaut adquiridos até este momento, imagine que precisamos implementar uma API REST para atualizar alguns dados de um autor em um 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 erro de validação devemos retornar o status HTTP 400;
* Em caso de autor não encontrado devemos retornar o status HTTP 404;

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, como você implementaria uma API REST com Micronaut para atualizar os dados do autor?

## O que seria bom ver nessa resposta?

* **Peso 5**: Criar um controller com um método para receber os dados do autor respeitando as restrições da atividade e atualizar o autor no banco de dados;
* **Peso 4**: Anotar o método do controller com @Put do Micronaut com uma URI usando o ID do autor para indicar o resource especifico;
* **Peso 1**: Usar o POSTman ou Insomnia para fazer os testes da API REST implementada;

## Resposta do Especialista:

* Crio uma classe de controller com um método para atualizar 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 @Put("/api/autores/{id}") para indicar que se trata de um endpoint para tratar uma requisição com verbo HTTP PUT, afinal geralmente utilizamos PUT para indicar atualização de recursos numa API REST;
* Crio uma classe de DTO pra receber o payload do autor no método do controller, anoto os atributos da classe com as anotações da Bean Validation para que o Micronaut consiga efetuar a validação antes de executar o método de fato além de retornar o status HTTP 400 em caso de erro;
* Uso o POSTman ou Insomnia para testar se o endpoint está respeitando as regras de validação do DTO;
* 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 retorno um status HTTP 404 indicando que o autor não foi encontrado no sistema;
* Faço mais um teste no POSTman passando um id inexistente no banco de dados para ter certeza que um status 404 seja retornado;
* Caso o autor exista no sistema, eu copio os dados do DTO para a entidade (aqui um método dentro da entidade pode ajudar), uso o repository para atualizar a entidade no 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;