

Logo da Orange Talents

# Recebendo dados via JSON

## Cenário:

Com o conhecimento de Micronaut para criar APIs REST, imagine que agora precisamos criar uma API REST para cadastrar no sistema um novo autor com seus respectivos livros. Para isso, um autor deve possuir as seguintes informações:

* nome;
* email;
* uma coleção de livros onde cada livro possui um título, um ISBN e uma data de publicação;

Não precisamos gravar em um banco de dados ou fazer qualquer validação em cima dos dados, mas devemos imprimir os dados da requisição no console e responder com status HTTP 201 contendo exatamente os mesmos dados da requisição.

Como você faria para implementar essa API REST?

## O que seria bom ver nessa resposta?

* **Peso 5**: Criação de um controller com um endpoint do tipo POST recebendo uma request com os dados do autor e sua coleção de livros, e retornando um status HTTP 201 com os dados da request no corpo da resposta;
* **Peso 2**: Uso da classe java.time.LocalDate para representar a data de publicação de um livro;
* **Peso 2**: Configurar Jackson para serializar objetos do tipo java.time.LocalDate no formato yyyy-MM-dd (pode ser via anotação, declarativa via application.yml ou programaticamente via Micronaut);
* **Peso 1**: Imprimir os dados no console usando a API de Logging (slf4j) em vez de um simples println();

## Resposta do Especialista:

* Crio uma nova classe de controller, anoto ela com @Controller, e crio um método adicionar() anotado com @Post para receber os dados da requisição na URI "/api/autores". Para os dados em JSON, eu crio um DTO NovoAutorRequest que recebe **nome** e **email**, ambos como String, e um atributo **livros** do tipo List, onde cada instância de livro também é um DTO, do tipo LivroRequest, contendo os atributos **titulo** e **isbn** como String, e um atributo **publicadoEm** como java.time.LocalDate. Por se tratar de DTOs, eu os crio como data class do Kotlin;
* Para imprimir os dados da requisição no console, eu uso a API de Logging do Slf4j que já vem junto com o Micronaut, criando uma instância de Logger no controller e logando os dados com a criticidade INFO;
* Para responder com status HTTP 201 contendo os mesmos dados da requisição eu faço o método adicionar() retornar uma HttpResponse.created(request). Lembrando que o método deve ter como retorno um HttpResponse<Any>; Aqui eu também poderia criar e retornar outro DTO como cópia da requisição, algo como NovoAutorResponse, mas por serem os mesmos dados e não haver transformações eu optei por retornar a própria request;
* Por fim, para garantir que o atributo **publicadoEm** do livro seja serializado em um formato legível (yyyy-MM-dd), eu configuro o Jackson com jackson.serialization.writeDatesAsTimestamps=false no arquivo application.yml da aplicação;