

Logo da Orange Talents

# Definindo headers e status customizados

## Cenário:

Imagine que precisamos implementar uma API REST que deve retornar um recibo de compra de ingresso pelo ID do ingresso comprado. Essa API vai ser consumida tanto um por aplicativo mobile como um navegador (browser). Contudo, por se tratar de um recibo na qual prioritariamente será impresso pelo usuário nossa API deve retorna o recibo no formato HTML em vez de JSON como estamos acostumados.

O rascunho do código seria algo parecido com este código abaixo:

@Controller  
class ImprimirReciboController(val repository: RecibosRepository) {  
  
 @Get("/api/recibos/{id}/html")  
 fun imprimir(@PathVariable id: UUID): HttpResponse<Any> {  
  
 val recibo = repository.findById(id)  
 if (recibo == null) {  
 return HttpResponse.notFound()  
 }  
  
 val html = ReciboResponse(recibo).toHtml() // converte DTO para HTML  
 return HttpResponse.ok(html)   
 }  
}

Assuma que código acima funciona, ele gera um HTML válido e devolve uma resposta válida para o browser, porém o conteúdo da API REST retornado é um JSON em vez do HTML. Por algum motivo o Micronaut está gerando uma resposta com formato JSON.

Como você ajustaria o código acima para que o Micronaut retornasse de fato um HTML em vez de JSON?

## O que seria bom ver nessa resposta?

* **Peso 7**: Definir o cabeçalho HTTP Content-Type como text/html na resposta da API REST usando Micronaut. Existem algumas maneiras simples de fazer isso, por exemplo, usando a anotação @Produces, ou indicando na anotação @Get(..., produces=TEXT\_XML) ou mesmo setando o cabeçalho diretamente na response via HttpResponse.ok(...).header("Content-Type", TEXT\_XML);
* **Peso 3**: Explicar como ele(a) faria o troubleshooing do problema antes de decidir o que fazer;

## Resposta do Especialista:

* Faria os testes manuais com POSTman ou Insominia para entender corretamente o problema, analisando a resposta HTTP recebida pelo POSTman juntamente com os cabeçalhos HTTP retornados;
* Ainda no POSTman, eu olharia o payload da resposta para confirmar de que se trata de um conteúdo JSON e também olharia o cabeçalho HTTP Content-Type retornado pelo servidor para verificar se veio um aplication/json em vez de um text/html. Com isso eu confirmaria de que o problema não está no lado cliente (browser) mas sim no lado servidor;
* Agora, dado que o conteúdo retornado pelo controller é um HTML, eu indicaria para o cliente (browser ou aplicativo mobile) de que se trata de um conteúdo HTML e não JSON. Para isso, eu defino o cabeçalho Content-Type do protocolo HTTP como text/html. No Micronaut, basta eu usar a anotação @Produces(MediaType.TEXT\_HTML) no método imprimir() indicando o tipo de conteúdo, no caso HTML;
* Por fim, faria mais um teste com POSTman para ter certeza de que a resposta foi retornada como esperada, tanto o conteúdo como o cabeçalho HTTP;