

# **Resposta REST**

**Status Codes** (código de Status)  $\Rightarrow$  São códigos que são enviados juntamente com a resposta de uma requisição HTTP. Esses códigos ajudam o cliente saber se a requisição foi bem sucedida. E se não foi, ajuda-o a entender qual foi a causa provável do erro.

**200** - Success

**500** - Server error

**404** - Not found



# AHHHH! 404...

Cockatoo can't find the page,  
but she can scream in a cup for you.

[Go back home.](#)

**Quem coloca essa página 404 geralmente é o desenvolvedor web** que está consumindo o REST API em questão. Assim como, toma outras decisões acerca de cada Status Codes.

Por isso, **é muito importante** que quando desenvolvermos os nossos REST APIs, usemos os **códigos apropriados** em cada resposta, sejam de sucesso ou de erro.

É muito fácil programar imaginando que tudo funcionará 100%, e que o usuário não cometerá erros, mas **um bom Web API deve considerar o máximo de cenários possíveis...**

de forma a dar códigos de **sucesso** quando tudo **ocorrer bem**, e também **prever possíveis erros**, e exibir os **códigos que melhor descrevem a causa do erro**.



## Formato dos Dados

O criador do REST API é quem decide em que formatos ele exibirá os dados como resposta. Como falado anteriormente, os **formatos mais comuns são JSON e XML.**

JSON é relativamente novo em comparação com o XML, mas devido a sua simplicidade, tem crescido muito mais. A maioria dos **REST APIs mais recentes, têm JSON como formato padrão ou principal.**

Em **muitos casos**, fornecem **ambos os formatos JSON e XML**. Nesse caso, ao fazer a requisição, o cliente pode informar qual o formato que ele deseja receber os dados.

Você pode estar se perguntando agora: como o cliente sabe quais os formatos disponíveis no servidor?

Uma observação: quando digo cliente, estou me referindo ao programador, ou ao desenvolvedor web que **consumirá** o Web API.

Respondendo a pergunta: o cliente descobre não apenas quais formatos de dados, mas quais recursos estão disponíveis através da **documentação do REST API.**

Apesar de não existir nenhuma obrigação de ter documentação, ou até mesmo um manual de como ela deve ser. É altamente **recomendável criar uma documentação, e seguir guias de boas práticas.**

E como o servidor identifica que tipo de formato de dados enviar? Através de um parâmetro do **Header**(ou cabeçalho), chamado **Content-Type** (Tipo de Conteúdo).



Junto com a requisição para acessar um Recurso, envia-se um **cabeçalho** contendo as informações necessárias. Essas informações necessárias, são chamadas de **Metadata** (ou metadados em português).

**Content-Type** é um tipo de Metadata ou Metadados, que indica qual o formato que se espera do servidor. Existem alguns padrões predefinidos de valores do Content-Type.

Para receber dados em XML, usa-se **text/xml**.  
Para receber dados em JSON, usa-se  
**application/json**.

Content-Type	application/json
Header	Value