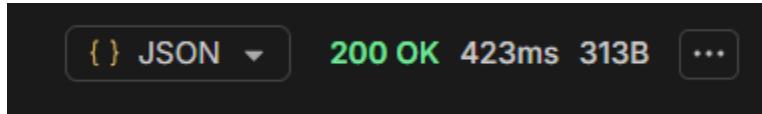


Requisições

1. Sucesso na operação.



2. Ao consultar um CEP inexistente o JSON retorna o status **error:true**. Pois nesse caso a requisição é válida mas não foi encontrada no banco de dados. A escolha da estratégia provavelmente foi feita por motivos de arquitetura e praticidade. O CEP inexistente não representa um erro técnico da requisição, pois sua formatação está correta. Quando se retorna status code 200 com um JSON padronizado simplifica o consumo de API, já que o cliente não precisa tratar exceções HTTP para um caso que faz parte do fluxo normal de uso.
3. Os dados XML são mostrados dentro da sua linguagem de marcação com um conjunto fixo de tags pré-definidas, que às vezes polui muito a visualização. Sendo o JSON uma forma mais prática.

A screenshot of a browser developer tools network tab. It shows a successful request with the status code 200 OK, a response time of 435ms, and a body size of 418B. The response type is set to XML. The response content is a multi-line XML document representing a CEP address.

```
Response Headers 13 >> XML 200 OK 435ms 418B ...
1   <?xml version="1.0" encoding="UTF-8"?>
2   <xmlcep>
3   <cep>30640-070</cep>
4   <logradouro>Avenida Afonso Vaz de Melo</logradouro>
5   <complemento>até 1999/2000</complemento>
6   <unidade></unidade>
7   <bairro>Barreiro</bairro>
8   <localidade>Belo Horizonte</localidade>
9   <uf>MG</uf>
10  <estado>Minas Gerais</estado>
11  <regiao>Sudeste</regiao>
12  <ibge>3106200</ibge>
13  <gia></gia>
14  <ddd>31</ddd>
15  <siafi>4123</siafi>
16  </xmlcep>
```

O Relatório do Engenheiro

1. O método usado foi o GET pois é o utilizado para fazer consultas dentro da API. O POST seria usado para alterar algo no servidor, enviar dados ou criar novos recursos.
2. Chave para o nome da cidade: **localidade** Chave para o nome da rua: **logradouro**