



Entendendo e Documentando RESTful APIs

Gerindo Erros (Exemplos)

Gerindo Erros (Exemplos)

Exemplo 01

Uma API possui um recurso chamado “**users**” que só tem disponível o verbo **GET**. Assim, fazemos a seguinte requisição e obtemos a resposta:

Gerindo Erros (Exemplos)

— — —

```
curl -i -X PUT http://example.com/users
```

```
HTTP/1.1 404 Not Found
```

Gerindo Erros (Exemplos)

Apesar do status code retornar algo que é verdade “**404 Not Found**”, essa resposta não é a mais indicada, visto que existe o código **405 Method Not Allowed**, que nesse caso se encaixa perfeitamente.

Gerindo Erros (Exemplos)

— — —

```
curl -i -X PUT http://example.com/users
```

```
HTTP/1.1 405 Method Not Allowed
```

Gerindo Erros (Exemplos)

Exemplo 02

O cliente solicita uma representação (XML por exemplo) que não está disponível no servidor. O código que devemos usar para uma representação não disponível deve ser o

406 Not Acceptable

Gerindo Erros (Exemplos)

— — —

```
curl -i http://example.com/users -H "Accept: xyz/abc"
```

```
HTTP/1.1 406 Not Acceptable
```


Gerindo Erros (Exemplos)

Exemplo 03

O cliente envia uma requisição **POST** com um **Content-Type** não suportado pelo servidor, o mesmo deve retornar o código

415 Unsupported Media Type

Gerindo Erros (Exemplos)

— — —

```
curl -X POST -i http://example.com/users \  
  -H "Content-Type: application/fake" \  
  -d 'Weirdly Formatted Data'
```

HTTP/1.1 415 Unsupported Media Type

Gerindo Erros (Exemplos)

Exemplo 04

Uma requisição **POST** é enviada ao servidor com um **JSON** mal formatado. O retorno do servidor deve conter o status code **400 Bad Request** e uma mensagem no corpo da resposta detalhando o erro.

Gerindo Erros (Exemplos)

Exemplo 04

O código **400 Bad Request** tem o propósito de indicar que o servidor não conseguiu entender a requisição por algum erro na sua sintaxe.

Gerindo Erros (Exemplos)

```
curl -X POST -i http://example.com/users \  
  -H "Content-Type: application/json" \  
  -d '{"name": "Jhon"}
```

HTTP/1.1 400 Bad Request

{"message": "757: unexpected token at '{\"name\": \"Jhon\"}'"}

Gerindo Erros (Exemplos)

Exemplo 05

Imagine uma requisição **POST** para criar um usuário em um site, mas, esse usuário já existe no BD. O que fazer?

Gerindo Erros (Exemplos)

Exemplo 05

O servidor deve retornar um código **409**
Conflict, que tem a finalidade de indicar ao
cliente que ele pode resolver um conflito e
tentar uma nova requisição.

Gerindo Erros (Exemplos)

```
curl -X POST -i http://example.com/users \  
  -H "Content-Type: application/json" \  
  -d '{"email": "jhon@example.com"}'
```

HTTP/1.1 409 Conflict

```
{"message": "Email jhon@example.com already exists on DB"}
```


Gerindo Erros (Exemplos)

Exemplo 06

Imagine uma requisição **GET** que tem o intuito de mostrar os dados de um determinado usuário, mas esse usuário não existe no BD, qual código devemos retornar?

Gerindo Erros (Exemplos)

Exemplo 06

Um código que tem esse propósito é o **404 Not Found**. Ele informa que nada foi encontrado em uma URI específica, ou seja, nesse caso, não existe retorno para a pesquisa.

Gerindo Erros (Exemplos)

— — —

```
curl -i http://example.com/users/jhon
```

```
HTTP/1.1 404 Not Found
```

Gerindo Erros (Exemplos)

Exemplo 07

Uma requisição DELETE é enviada ao servidor para excluir um usuário, qual código devemos retornar? **204 No Content**

Gerindo Erros (Exemplos)

Exemplo 07

O código **204 No Content** indica que a requisição foi completada com sucesso, mas que não existe nenhum conteúdo adicional para ser enviado para o cliente.

Gerindo Erros (Exemplos)

— — —

```
curl -i -X DELETE http://example.com/users/jhon
```

```
HTTP/1.1 204 No Content
```

Gerindo Erros (Exemplos)

Exemplo 08

Após apagar um usuário, o cliente volta a requisitar o mesmo com uma requisição GET. Nesse caso, o mesmo não existe mais, qual código devemos retornar?

Gerindo Erros (Exemplos)

Exemplo 08

Inicialmente podemos usar o código **404 Not Found**, mas ele é muito genérico e caso a aplicação, do lado do servidor, consiga identificar quem aquele usuário solicitado já existiu no BD, podemos então retornar o código **410 Gone**.

Gerindo Erros (Exemplos)

Exemplo 08

O código **410 Gone** trás a ideia de que aquele recurso já existia e agora não existe mais.

Gerindo Erros (Exemplos)

— — —

```
curl -i -X DELETE http://example.com/users/jhon
```

HTTP/1.1 204 No Content

```
curl -i http://example.com/users/jhon
```

HTTP/1.1 410 Gone