



# **Entendendo e Documentando RESTful APIs**

# Analizando uma resposta HTTP

# Analizando uma resposta HTTP

---

**Nossa primeira requisição com o cURL**

```
curl -i https://jsonplaceholder.typicode.com/posts/1
```

# Analizando uma resposta HTTP

---

O que o cURL mostra como resposta de uma requisição HTTP pode ser dividido em 4 partes:

# Analizando uma resposta HTTP

---

- 1- **Start-Line** (Linha de início / Obrigatória)
- 2- **Header Fields** (Cabeçalho de Campos / Pode ter 0 ou mais)
- 3- **Empty Line** (Linha em branco / Obrigatória)
- 4 - **Message-Body** (Corpo da mensagem / Opcional)

# Analizando uma resposta HTTP

— — —

Apesar de todas as requisições terem respostas diferentes, pelo menos a linha inicial e a linha em branco existirão.

# Analizando uma resposta HTTP

```
jacksonpires:~/workspace $ curl -i https://jsonplaceholder.typicode.com/posts/1
HTTP/1.1 200 OK
Date: Wed, 21 Sep 2016 12:30:52 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 292
Connection: keep-alive
Set-Cookie: __cfduid=d8587bd00c2dde9abfbae2c3d5b4fa9c61474461052; expires=Thu, 21-Sep-17 12:30:52 GMT; path=/; domain=.typicode.com; HttpOnly
X-Powered-By: Express
Vary: Origin, Accept-Encoding
Access-Control-Allow-Credentials: true
Cache-Control: public, max-age=14400
Pragma: no-cache
Expires: Wed, 21 Sep 2016 16:30:52 GMT
X-Content-Type-Options: nosniff
Etag: W/"124-yv65LoT2uMHRpn06wNpAcQ"
Via: 1.1 vegur
CF-Cache-Status: HIT
Server: cloudflare-nginx
CF-RAY: 2e5d87e898792561-ORD

{
  "userId": 1,
  "id": 1,
  "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
  "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
}
```

# Analizando uma resposta HTTP

---

## 1 - Start-Line

A **Start-Line** pode ser dividida em duas partes, **Request-Line** e **Status-Line**, onde, na imagem, **HTTP/1.1** é a **Request-Line** que indica a versão do HTTP que foi usada, e o **200 OK** que é a **Status-Line** representando que houve uma resposta.

**HTTP/1.1 200 OK**



# Analizando uma resposta HTTP

---

## 2 - Header Fields

Os Header Fields representam os metadados da requisição e resposta HTTP. Eles contém informações sobre como a transferência dos dados deve ser manipulada.

```
Date: Wed, 21 Sep 2016 12:30:52 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 292
Connection: keep-alive
Set-Cookie: __cfduid=d8587bd00c2dde9abfbae2c3d5b4fa9c614
path=/; domain=.typicode.com; HttpOnly
X-Powered-By: Express
Vary: Origin, Accept-Encoding
Access-Control-Allow-Credentials: true
Cache-Control: public, max-age=14400
Pragma: no-cache
Expires: Wed, 21 Sep 2016 16:30:52 GMT
X-Content-Type-Options: nosniff
Etag: W/"124-yv65LoT2uMHRpn06wNpAcQ"
Via: 1.1 vegur
CF-Cache-Status: HIT
Server: cloudflare-nginx
CF-RAY: 2e5d87e898792561-ORD
```

# Analizando uma resposta HTTP

---

## 2 - Header Fields

Nesta resposta podemos notar alguns header fields, como por exemplo:

**Content-Type**

**Content-Length**

**X-Powered-By**

```
Date: Wed, 21 Sep 2016 12:30:52 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 292
Connection: keep-alive
Set-Cookie: __cfduid=d8587bd00c2dde9abfbae2c3d5b4fa9c614
path=/; domain=.typicode.com; HttpOnly
X-Powered-By: Express
Vary: Origin, Accept-Encoding
Access-Control-Allow-Credentials: true
Cache-Control: public, max-age=14400
Pragma: no-cache
Expires: Wed, 21 Sep 2016 16:30:52 GMT
X-Content-Type-Options: nosniff
Etag: W/"124-yv65LoT2uMHRpn06wNpAcQ"
Via: 1.1 vegur
CF-Cache-Status: HIT
Server: cloudflare-nginx
CF-RAY: 2e5d87e898792561-ORD
```

# Analizando uma resposta HTTP

— — —

## 2 - Header Fields

**Content-Type:** Informa como a representação é serializada.

**Content-Length:** Informa o tamanho do corpo da mensagem e é indicado em octetos.

# Analizando uma resposta HTTP

---

## 2 - Header Fields

**X-Powered-By:** Header Fields não oficiais, por convenção começam com um **X**. No entanto essa prática está em desuso pois existem diversos Header Fields oficiais e devemos tentar usá-los antes de criar algum.

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers>

Mesmo assim, ao se criar um novo Header, atualmente, indica-se não começar mais com o **X**.


<http://stackoverflow.com/questions/3561381/custom-http-headers-naming-conventions>

# Analizando uma resposta HTTP

---

## 3 - Empty Line

A linha em branco serve apenas para delimitar o fim dos Header Fields e o início do corpo da mensagem.



```
Vary: Origin, Accept-Encoding
Access-Control-Allow-Credentials: true
Cache-Control: public, max-age=14400
Pragma: no-cache
Expires: Wed, 21 Sep 2016 16:30:52 GMT
X-Content-Type-Options: nosniff
Etag: W/"124-yv65LoT2uMHRpn06wNpAcQ"
Via: 1.1 vegur
CF-Cache-Status: HIT
Server: cloudflare-nginx
CF-RAY: 2e5d87e898792561-ORD

{
  "userId": 1,
  "id": 1,
  "title": "sunt aut facere repellat provident occaecati
  "body": "quia et suscipit\nsuscipit recusandae consequ
ut ut quas totam\nnostrum rerum est autem sunt rem eveni
```

# Analizando uma resposta HTTP

---

## 4 - Message-Body

O corpo da mensagem contém os dados que foram enviados do servidor em resposta à requisição feita. Nesse caso temos um **JSON**.

```
{
  "userId": 1,
  "id": 1,
  "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
  "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
}
```

jacksonpires:~/workspace \$

# Analizando uma resposta HTTP

---

Uma outra forma de fazer uma requisição é usando a opção **-v**, que faz com que o resultado seja mais verboso, mostrando de fato como ocorreu a requisição.

```
curl -v -i https://jsonplaceholder.typicode.com/posts/1
```