

# ≡ Introdução

- REST = Representational State Transfer
- Basicamente refere-se a transferência de uma **representação** de um recurso (hospedado no servidor) para o cliente.
- É um tipo de arquitetura de software que segue princípios para desenvolver sistemas mais eficientes visando escalabilidade.

---

## ◆ Princípios da arquitetura RESTful

- **Interface uniforme:** a api deve ser consistente seguindo uma padronização.
- **Client/server:** segue arquitetura cliente-servidor.
- **Stateless:** não mantém informações sobre o estado da conexão. Nesse caso o cliente deve manter o estado da sessão. Cada requisição é autocontida, isto é, o estado não é carregado entre requisições.
- **Cacheable:** permitir que respostas do servidor sejam salvas por um período de tempo.
- **Sistemas em camadas:** o servidor possui diferentes camadas de serviços, sendo transparente para o cliente.
- **Código sob-demanda:** o servidor REST pode enviar código para o cliente executar.

---

## ◆ Anatomia de uma URI

<https://api.example.com/1/weather/UK/London/2019-06-19?units=metric>

Protocol	https://
Domain	api.example.com
Version	1.0
API Endpoint	/weather/
Entity	UK/London
Record	2019-06-19
Parameters	?units=metric

---

## ◆ URI e URL

**URI** Uniform Resource Identifier

identifica um recurso como um livro, um produto, uma pessoa...

**URL** Uniform Resource Locator

localiza um recurso - de modo que seja possível acessar o recurso

---

## ◆ Verbos

Request: quando o cliente envia uma mensagem para o servidor.

APIs REST seguem padronização dos verbos **HTTP**:

- POST - usado para **criar** recursos (**C**)
- GET - para **ler** recursos (**R**)
- PUT - usado para **atualizar** um recurso existente (**U**)
- PATCH: semelhante ao POST e PUT, com objetivo de fazer updates parciais num recurso.

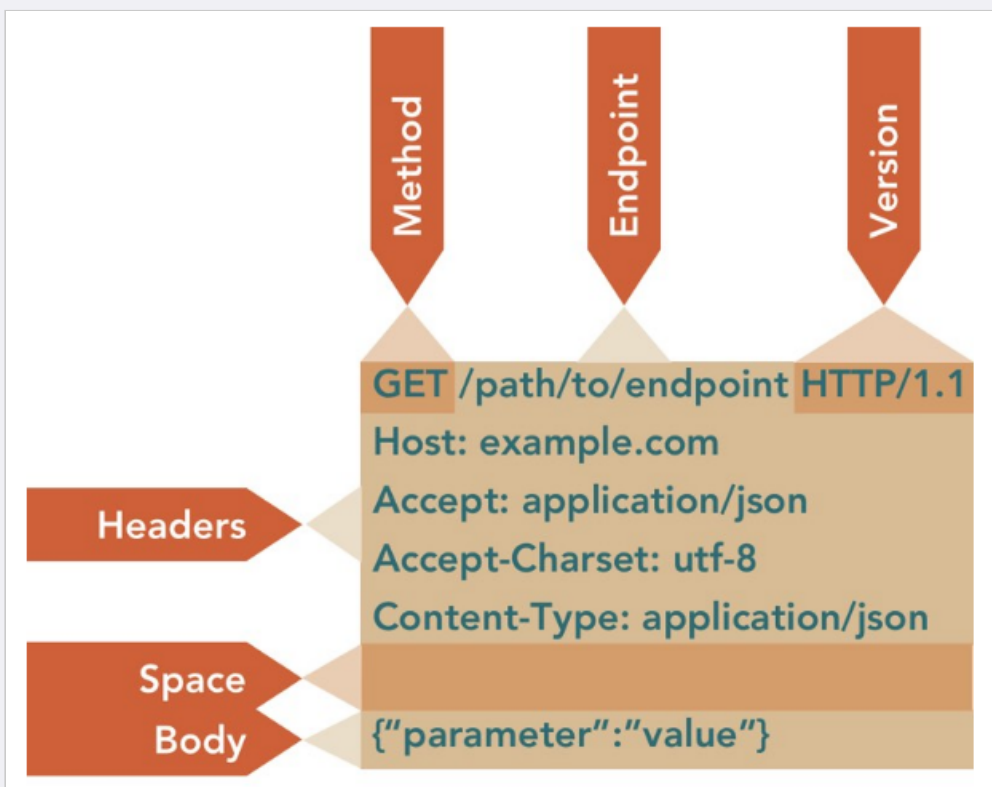
- DELETE - **deletar** um recurso (D)

---

## ◆ Request

Uma requisição HTTP contém:

- um método de requisição, URL de requisição, **campos do cabeçalho**, corpo da mensagem



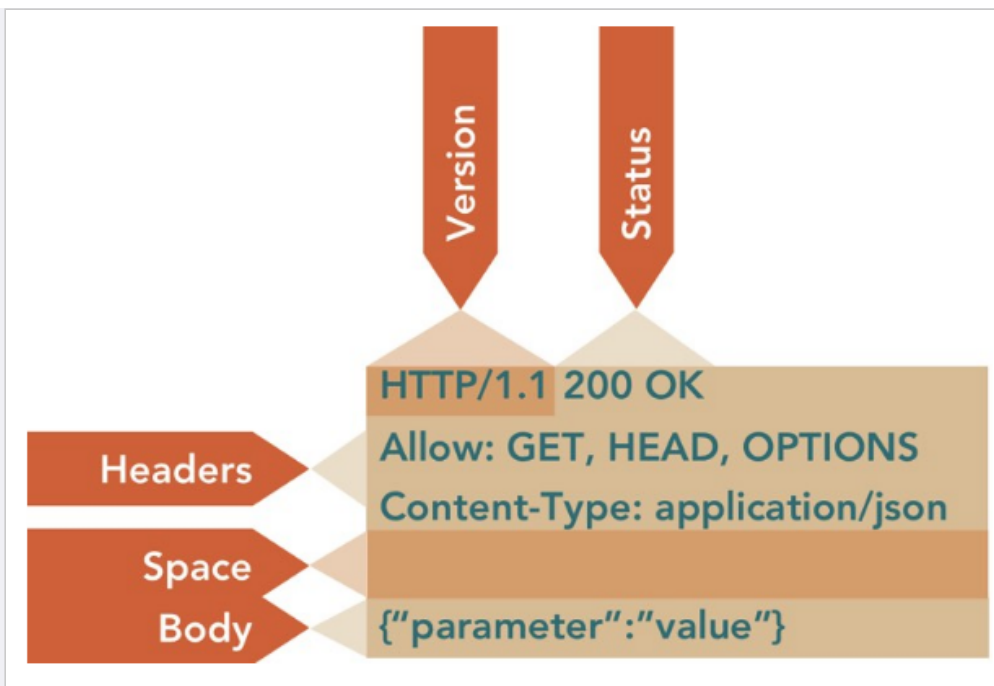
---

## ◆ Response

Quando o servidor retorna uma resposta ao cliente.

Uma mensagem HTTP response contém:

- código de resultado
- campos de cabeçalho
- corpo de mensagem.



---

## ◆ Códigos de retorno do servidor

Lista de [código](#)

- 200: OK - sucesso
- 201: created - sucesso ao criar o recurso
- 401: indica que a requisição precisa ser autenticada.
- 404: indica que o recurso requisitado não está disponível
- 500: indica a ocorrência de um erro interno do servidor web.
- 503: servidor está sobrecarregado e não pode atender a requisição.