

REST Cheat Sheets

Algumas boas práticas são:

- Utilizar **substantivos** para nomear os recursos e não verbos.
Os verbos devem ser utilizados para ações (*Actions*);
- Recomenda-se também adotar os nomes dos recursos no plural
(Apesar que boas APIs, como LinkedIn, utilizam no singular);
- Utilizar letras minúsculas;
- Se necessário utilizar separador, utilize hífen para melhor legibilidade.

Depois da definição dos recursos, é necessário identificar quais **ações** se aplicam a eles e como serão mapeados para a API. Princípios RESTful fornecem estratégias para lidar com ações CRUD usando métodos HTTP mapeados, tal como:

- `GET /tickets` Retorna a lista de bilhetes
- `GET /tickets/12` Retorna um bilhete específico
- `POST /tickets` Cria um novo bilhete
- `PUT /tickets/12` Atualiza o bilhete #12
- `PATCH /tickets/12` Atualiza parcialmente o bilhete #12
- `DELETE /tickets/12` Apaga o bilhete #12

Se uma relação só pode existir dentro de outro recurso, os princípios RESTful fornecem orientações úteis. Tomando por base um cenário hipotético, um bilhete consiste em um certo número de mensagens. Estas mensagens podem ser logicamente mapeadas para o endpoint `/tickets`, como em:

- `GET /tickets/12/messages` Retorna a lista de mensagens do bilhete #12
- `GET /tickets/12/messages/5` Retorna a mensagem #5 do bilhete #12
- `POST /tickets/12/messages` Cria uma nova mensagem no bilhete #12
- `PUT /tickets/12/messages/5` Atualiza a mensagem #5 do bilhete #12
- `PATCH /tickets/12/messages/5` Atualiza parcialmente a mensagem #5 do bilhete #12
- `DELETE /tickets/12/messages/5` Apaga a mensagem #5 do bilhete #12

Aplique corretamente o HTTP Status Code

O HTTP Status Code é a forma que seus serviços irão retornar as respostas para seus clientes. É importante aplicar corretamente o HTTP Status Code para cada operação, veja abaixo como podemos retornar corretamente o HTTP Status Code para cada Verbo HTTP.

- **GET:**
 - Retorne 200(OK) para caso de sucesso
 - Retorne 404 (NOT FOUND) se a entidade não for encontrada
- **POST:**
 - Retorne 201 (CREATED) para caso um novo recurso seja criado com sucesso
 - Retorne 400 (BAD REQUEST) caso a solicitação contenha dados inválidos
 - Retorne 422 (Unprocessable Entity) caso a solicitação caia em alguma regra de negócio
- **PUT:**
 - Retorne 200 (OK) se for atualizar um recurso existente
 - Retorne 400 (BAD REQUEST) caso a solicitação contenha dados inválidos
 - Considere utilizar 409 (CONFLICT) caso não consiga atualizar um recurso existente
- **DELETE:**
 - Retorne 204 (No Content) para sucesso
 - Retorne 404 (NOT FOUND) se a entidade não for encontrada