



Entendendo e Documentando RESTful API's

O que é REST?

O que é REST?

“O termo transferência de estado representacional – **REST (Representational State Transfer)** foi introduzido e definido no ano de **2000** através de uma **tese de Ph.D** do cientista **Roy Fielding**, um dos **principais autores** da especificação do protocolo **HTTP**”

O que é REST?

“O intuito geral da tese era a **formalização**
de um conjunto de **melhores práticas**
denominadas **constraints**”

O que é REST?

“Essas **constraints** tinham como objetivo determinar a forma na qual padrões como **HTTP** e **URI** deveriam ser modelados, aproveitando de fato todos os recursos oferecidos pelos mesmos.”

Constraints

Constraints

— — —

Cliente-Servidor

Constraints / Cliente-Servidor

“A principal característica dessa constraint é **separar as responsabilidades** de diferentes partes de um sistema.”

Constraints / Cliente-Servidor

“Essa divisão pode se dar de diversas formas, iniciando por exemplo com uma separação entre mecanismos da **interface do usuário** e o **back-end** da aplicação.”

Constraints / Cliente-Servidor

“Isso nos permite a evolução e **escalabilidade**
destas responsabilidades **de forma**
independente.”

Constraints

— — —

Stateless

Constraints / Stateless

“Essa característica propõe que **cada requisição** ao servidor **não deve ter ligação com requisições anteriores ou futuras**, ou seja, cada requisição deve conter todas as informações necessárias para que ela seja tratada com sucesso pelo servidor.”

Constraints

— — —

Cache

Constraints / Cache

“Para uma melhor performance, um sistema **REST** deve **permitir** que suas respostas sejam passíveis de **cache**.”

Constraints

— — —

Interface Uniforme

Constraints / Interface Uniforme

“Bastante esforço deve ser feito para que o sistema possua uma **interface modelada** seguindo alguns padrões importantes.”

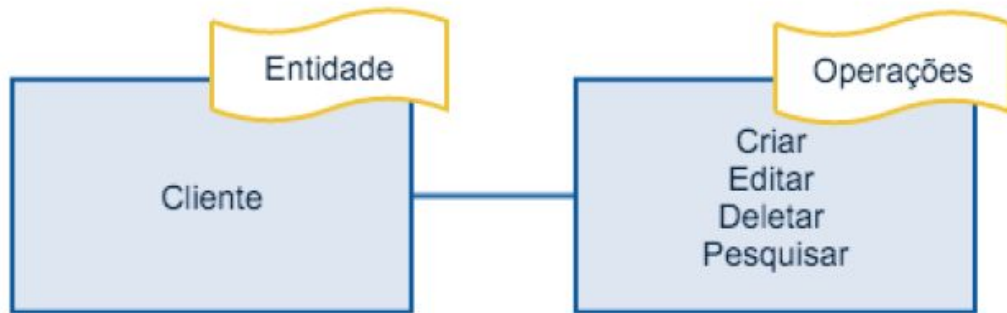
Constraints / Interface Uniforme

Quando se fala sobre uma interface, os elementos abaixo devem ser considerados:

- **Recursos**
- **Mensagens autodescritivas**
- **Hypermedia**

Constraints / Interface Uniforme

“Devemos pensar em criar uma interface que permita a manipulação desses conceitos. Veja o exemplo:”



Constraints

— — —

Sistema em Camadas

Constraints / Sistema em Camadas

“Com o intuito de permitir a escalabilidade necessária para grandes sistemas distribuídos, um sistema REST deve ter a capacidade de adicionar elementos intermediários e que sejam totalmente transparentes para seus clientes.”

Ex: Balanceador de Carga

Constraints

— — —

Código Sob Demanda

Constraints / Código Sob Demanda

“Código sob demanda é a única constraint REST
opcional”

Constraints / Código Sob Demanda

“A ideia é aumentar a flexibilidade dos clientes, como por exemplo um código javascript que só é baixado quando uma determinada página é carregada”

Constraints / Código Sob Demanda

“Apesar de ser algo interessante, essa prática **reduz a visibilidade**, por isso essa é uma prática opcional.”