

targettrust
treinamento e tecnologia



Riguel Figueiró

Consultor de software na ThoughtWorks
Apaixonado por metodologias ágeis,
integração contínua, qualidade e disseminação
e compartilhamento de conhecimento!

Como falar comigo:

- riguel@dkosoftware.com.br
- www.linkedin.com/in/riguel-figueiro/
- github.com/riguelbf





{REST}

**RESTful
Web Services**



O que é o Rest – Em poucas palavras

Rest é ...

- *Sobre recursos e como representar recursos em jeitos diferentes.*
- *REST é sobre comunicação cliente-servidor.*
- *REST oferece uma maneira simples, interoperável e flexível de escrevendo serviços web que podem ser muito diferentes de outras técnicas.*
- *Vem do estudo da tese de Roy Fielding.*



E o não que é o Rest

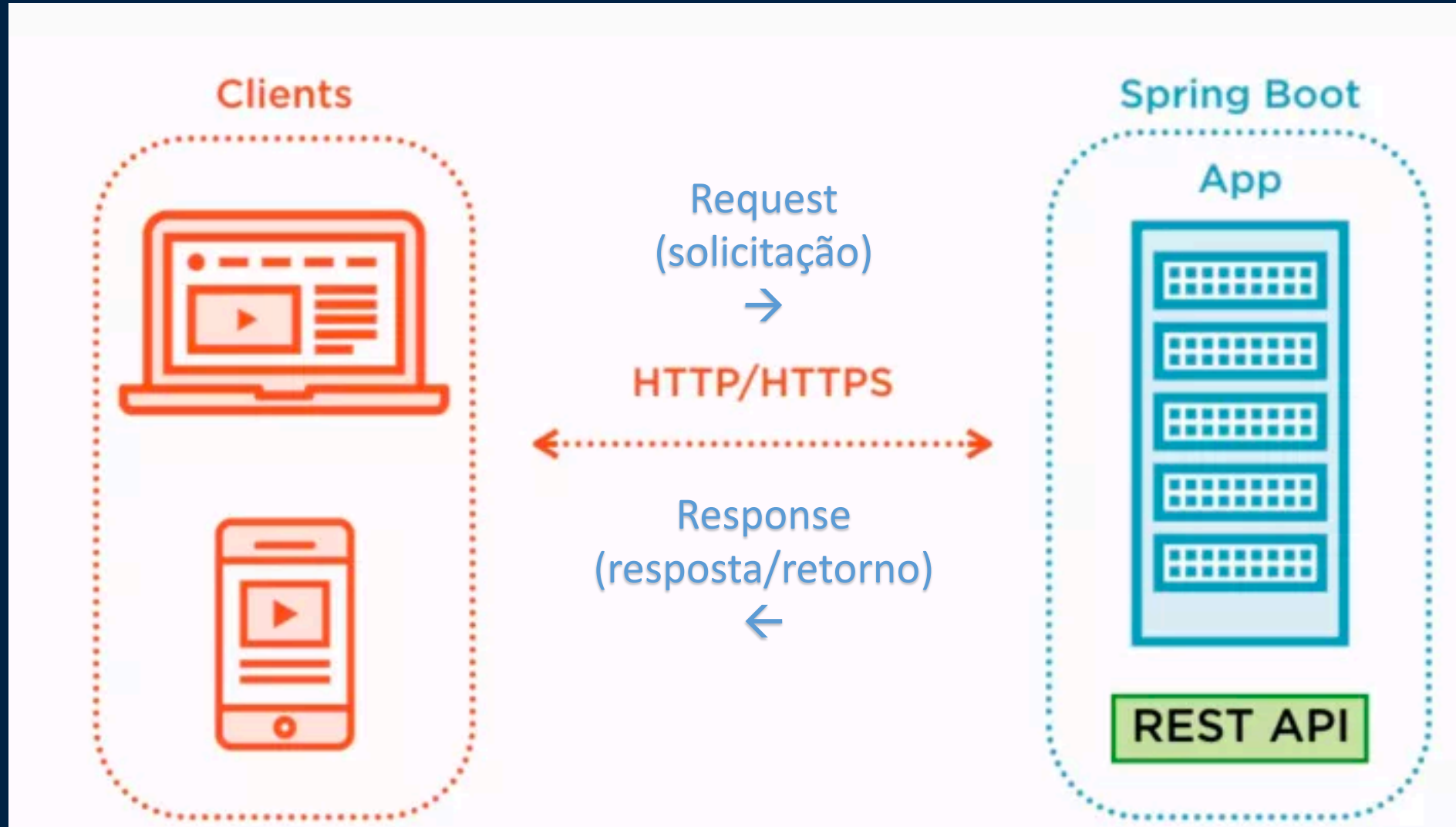
Rest NÃO é ...

- *Um protocolo.*
- *Standard e/ou um padrão.*
- *Um substituto para o SOAP.*

“Então pra eu aprender Gradle eu preciso saber Groovy?”

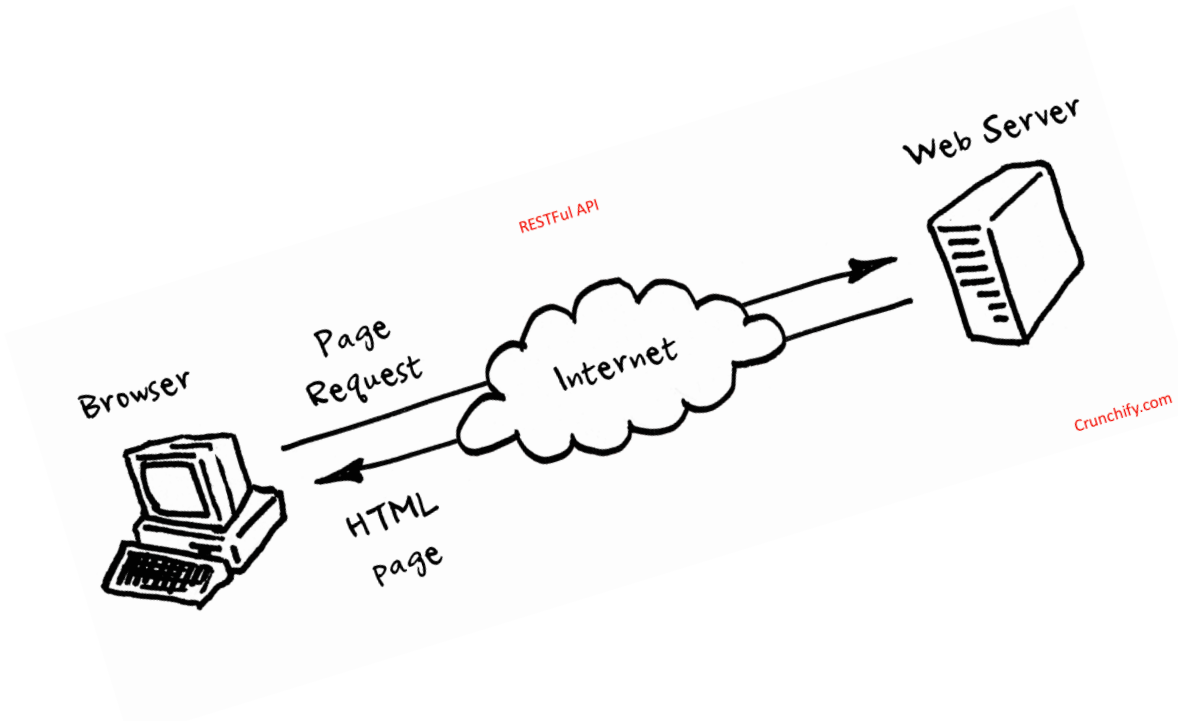
- Um estilo arquitetônico
- REST é a arquitetura da Web como funciona hoje e, então já é usado na web!
- É um modelo de arquitetura de software que é usado para descrever sistemas distribuídos como.
- Foi desenvolvido em paralelo com o protocolo HTTP.

A Web



Rest

1. *O cliente solicita um recurso específico do servidor.*
2. *O servidor responde a essa solicitação entregando o recurso solicitado.*
3. *O servidor não possui informações sobre nenhum cliente.*
4. *Portanto, não há diferença entre as duas solicitações do mesmo cliente.*
5. *Um modelo em que as representações dos recursos são transferidas entre o cliente e o servidor.*



GET and POST Request

Requests

- REQUEST

GET /news/ HTTP/1.1

Host: example.org Accept-Encoding: compress, gzip

User-Agent: Python-httpplib2

Aqui é um GET request para

«http://example.org/news/»

Method = GET



Response

- Response

HTTP/1.1 200 Ok Date: Thu, 07 Aug 2008

15:06:24 GMT Server: Apache ETag:

"85a1b765e8c01dbf872651d7a5" Content-Type:

text/html Cache-Control: max-age=3600

.....

Method = GET



Requests & Responses

- Um estilo arquitetônico
- A solicitação é para um recurso identificado por um URI
- (URI = Identificador de Recurso Unificado).
- Neste caso, o recurso é «<http://example.org/news/>»
- Recursos ou endereçamento são muito importantes.
- Todo recurso é endereçável por URL.
- Para alterar o estado do sistema, basta alterar um recurso.

UR exemplos ...

- <http://localhost:9999/restapi/books>
 - **GET** – obtem lista de books
 - **POST**- adiciona/inclui um novo livro
- <http://localhost:9999/restapi/books/{id}>
 - **GET** – get the book whose id is provided
 - **POST** – atualiza um livro com o ID fornecido
 - **DELETE** – exclui um livro com o ID fornecido

Métodos HTTP

Safety / Idempotency

HTTP Method	Safe	Idempotent
GET	✓	✓
PUT	✗	✓
DELETE	✗	✓
POST	✗	✗
PATCH	✗	✗



CREATE



READ



UPDATE



DELETE

C R U D

Operações CRUD mapeadas para HTTP

Métodos em Serviços Web RESTful

OPERATION	HTTP METHOD
Create	POST
Read	GET
Update	PUT or POST
Delete	DELETE

HTTP Status Codes

Level 200 (Success)

200 : OK

201 : Created

203 : Non-Authoritative
Information

204 : No Content

Level 400

400 : Bad Request

401 : Unauthorized

403 : Forbidden

404 : Not Found

409 : Conflict

Level 500

500 : Internal Server Error

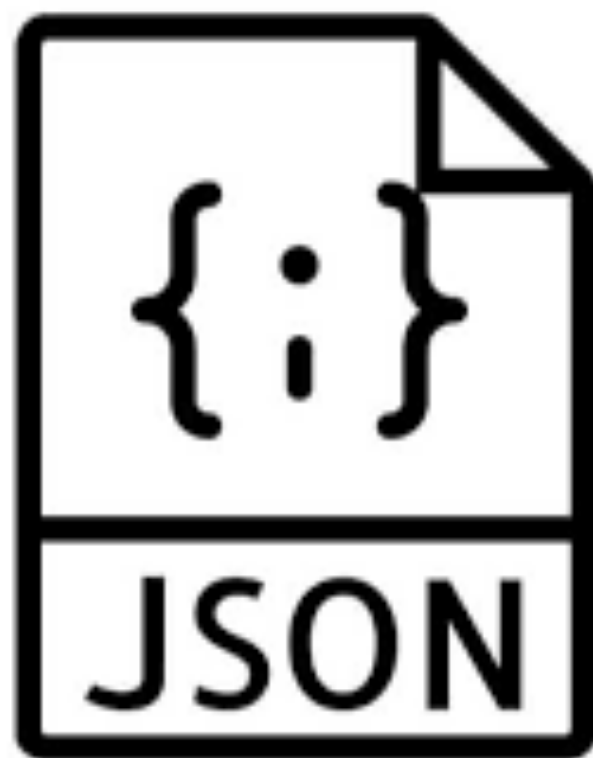
503 : Service Unavailable

501 : Not Implemented

504 : Gateway Timeout

599 : Network timeout

502 : Bad Gateway





POSTMAN



Pergunta

Façam o máximo de perguntas que desejarem!!