Fundamentos de DevOps

Aula 06 - Introdução ao Desenvolvimento de Web APIs

Prof. Esp. Guilherme Jorge Aragão da Cruz

guilherme.cruz@alumni.usp.br

linkedin.com/in/guijac

Roteiro

- O JavaScript Object Notation (JSON);
- O Fluxo de Mensagens via HTTP;
- Verbos/Métodos HTTP
  - GET;
  - POST;
  - PUT;
  - DELETE;
- Juntando as Peças.
- Application Programming Interface (APIs);
- REST e RESTfull;
- O Framework Spring Boot;
- Testando REST APIs com Postman;
- Referências Bibliográficas.

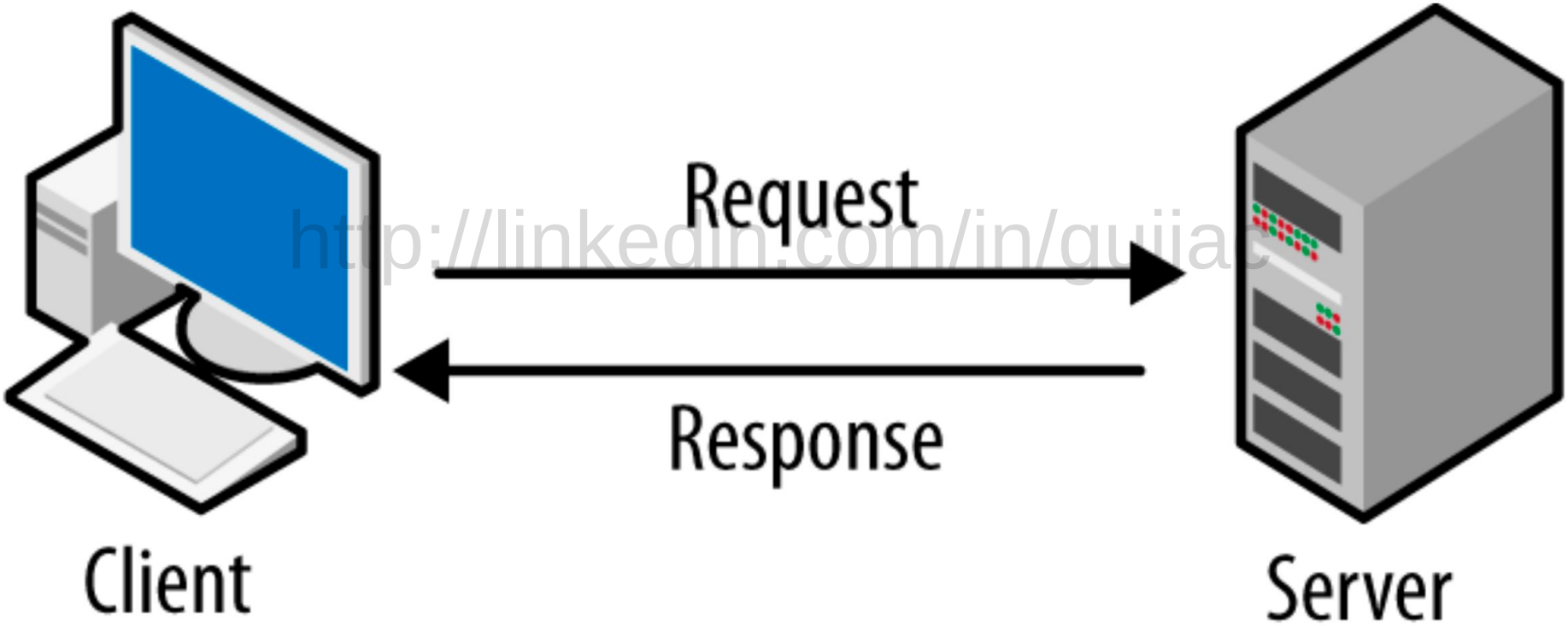
O JavaScript Object Notation (JSON)

- Arquivo **compacto**, de **padrão aberto**, para troca de dados simples e rápida entre sistemas, especificado por Douglas Crockford em 2000, que utiliza **texto legível a humanos**, no formato **atributo-valor**;
- O tipo de mídia da Internet oficial (MIME) para o JSON é **application/json** e nomes de arquivos JSON usam a extensão **.json**.


```
{
    "firstName" : "Guilherme",
    "id" : 1,
    "lastName" : "Cruz"
}
```

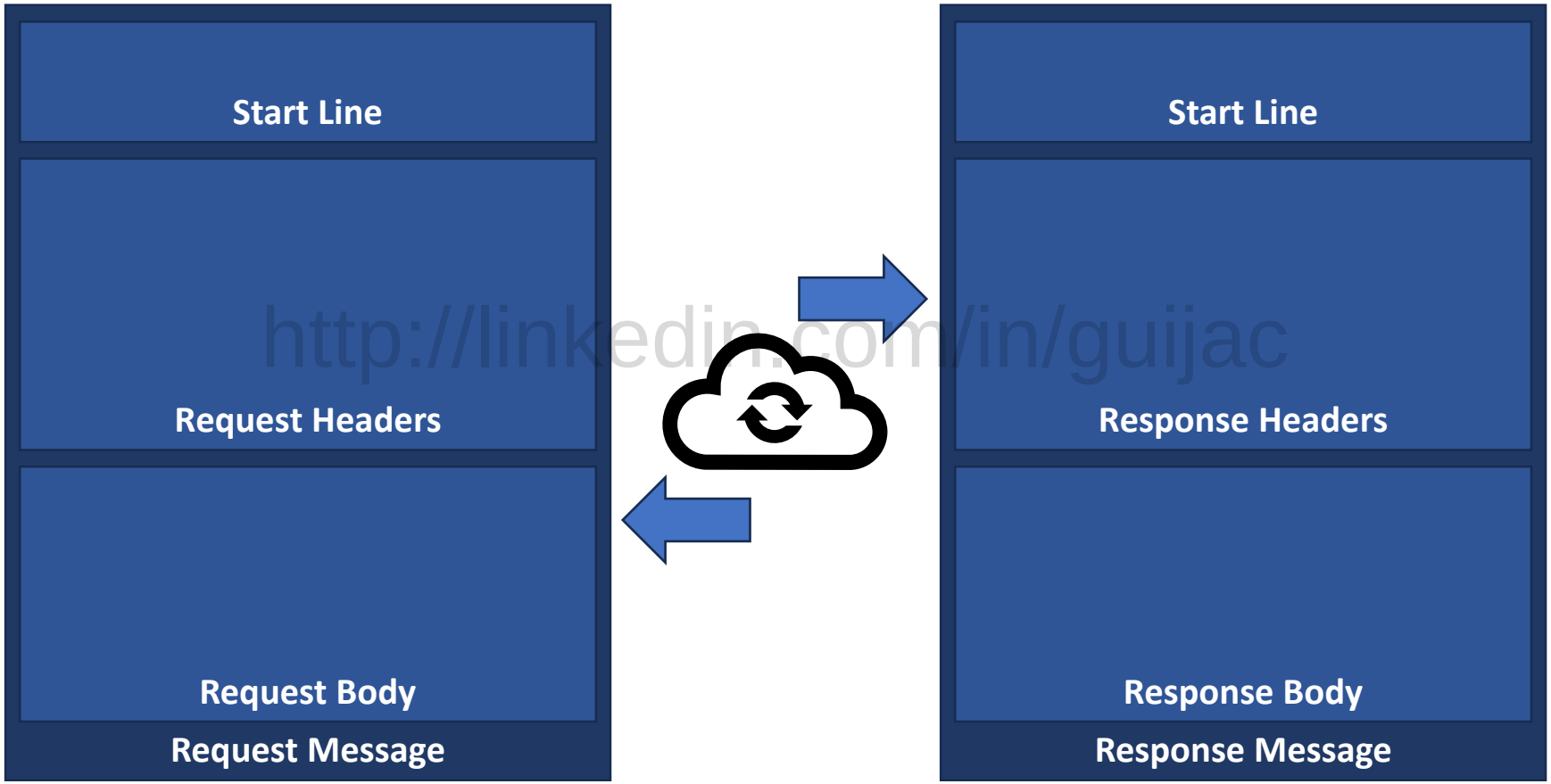
Simples arquivo JSON de um objeto “Student”, com os atributos “id”, “firstName” e “lastName”.

O Fluxo de Mensagens via HTTP



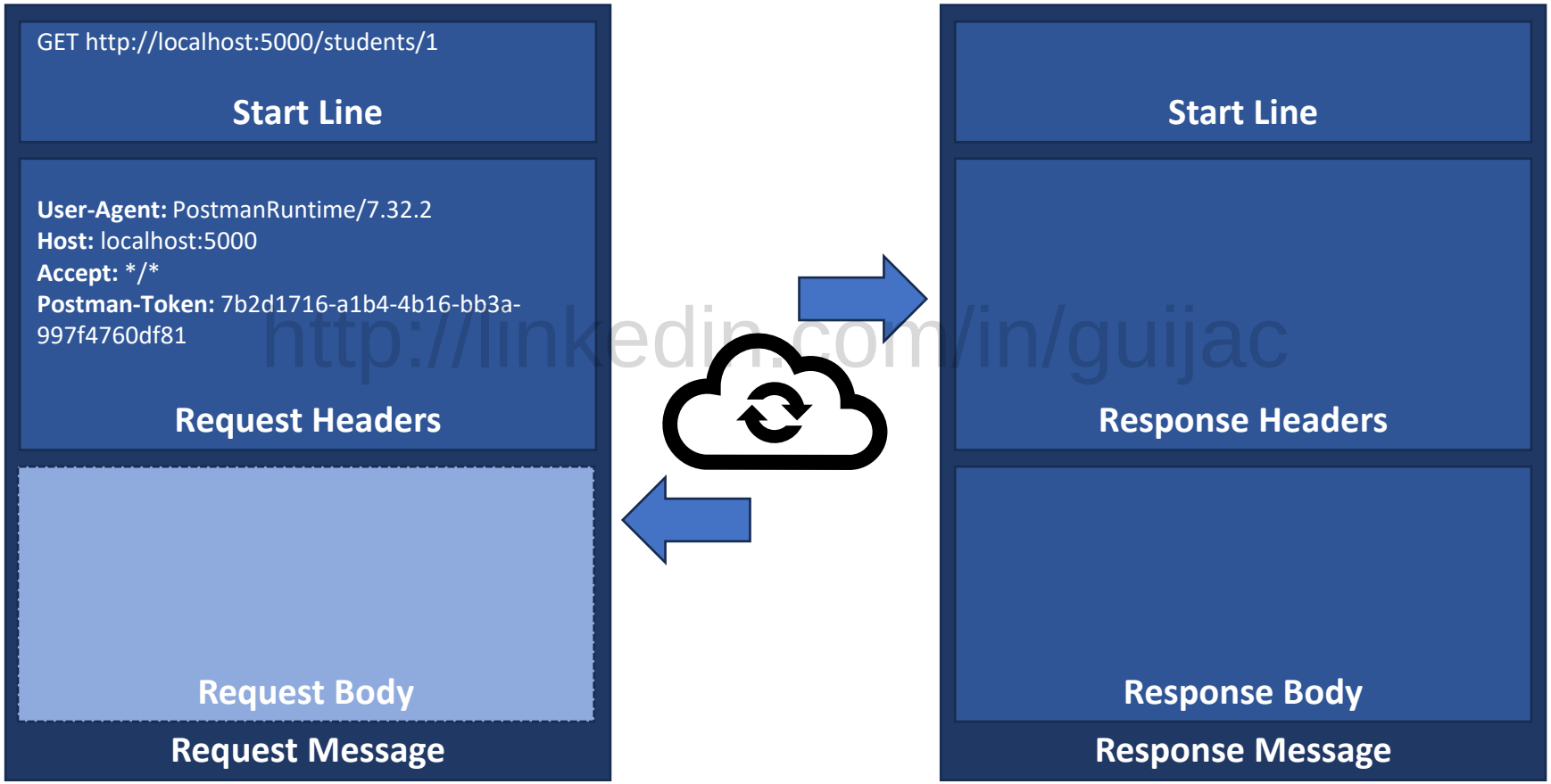
Fonte: Client-Server Architecture | EN.601.421: Object-Oriented Software Engineering (OOSE) (darvishdarab.github.io)

O Fluxo de Mensagens via HTTP



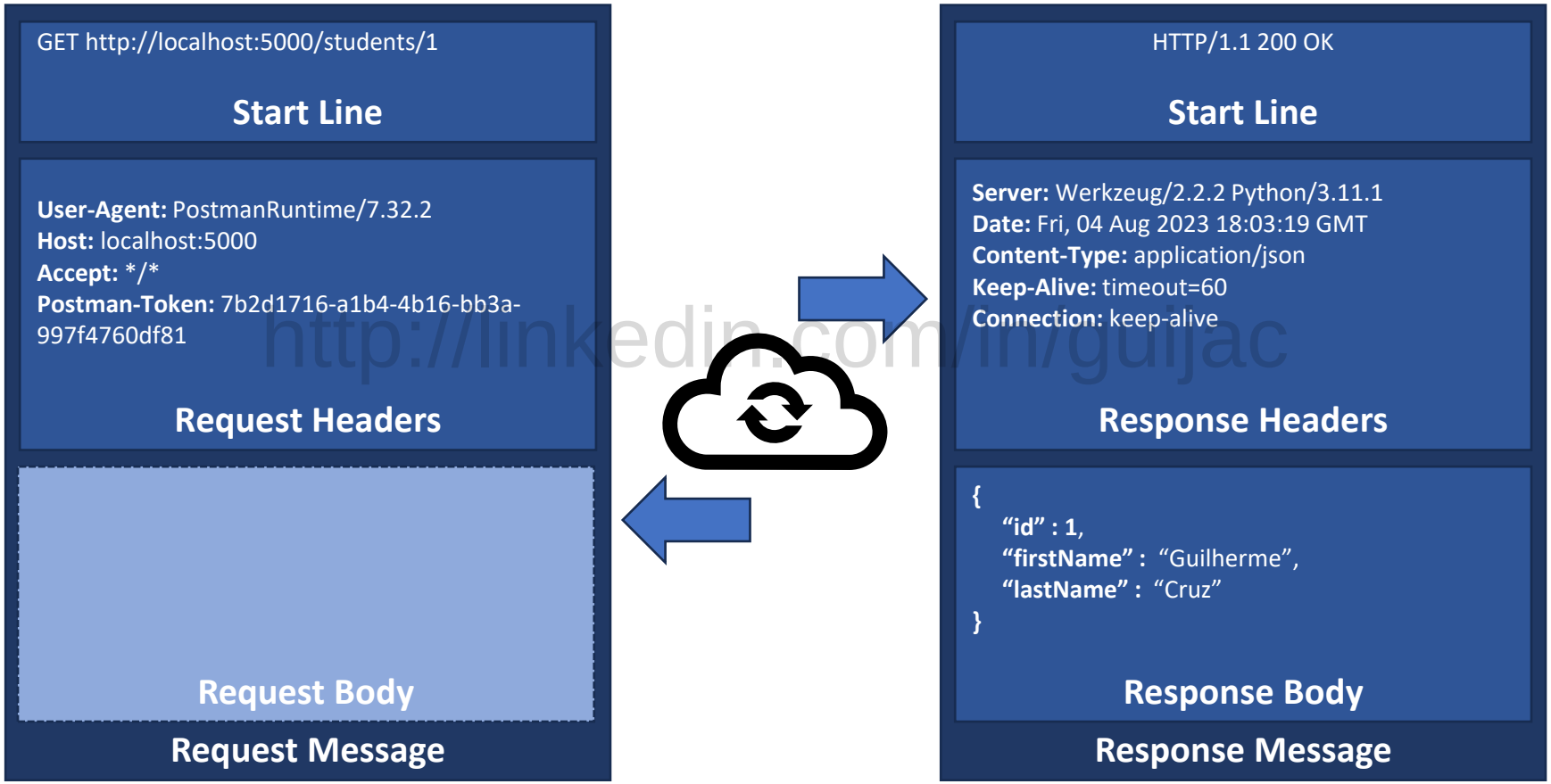
Fluxo simples de troca de mensagens com Protocolo HTTP.

Verbos/Métodos HTTP: GET



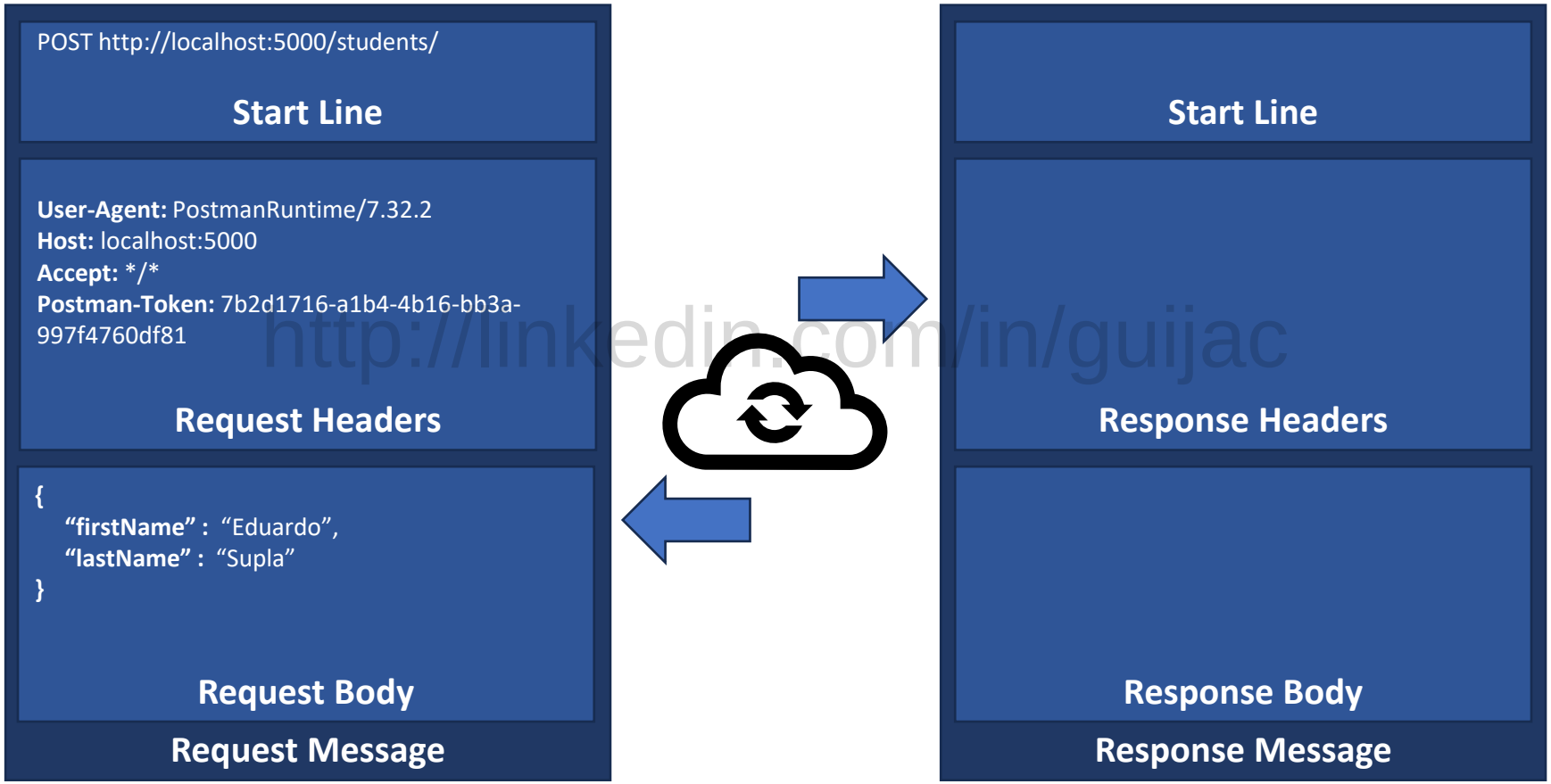
Fluxo simples de troca de mensagens com Protocolo HTTP utilizando verbo/método “GET”

Verbos/Métodos HTTP: GET



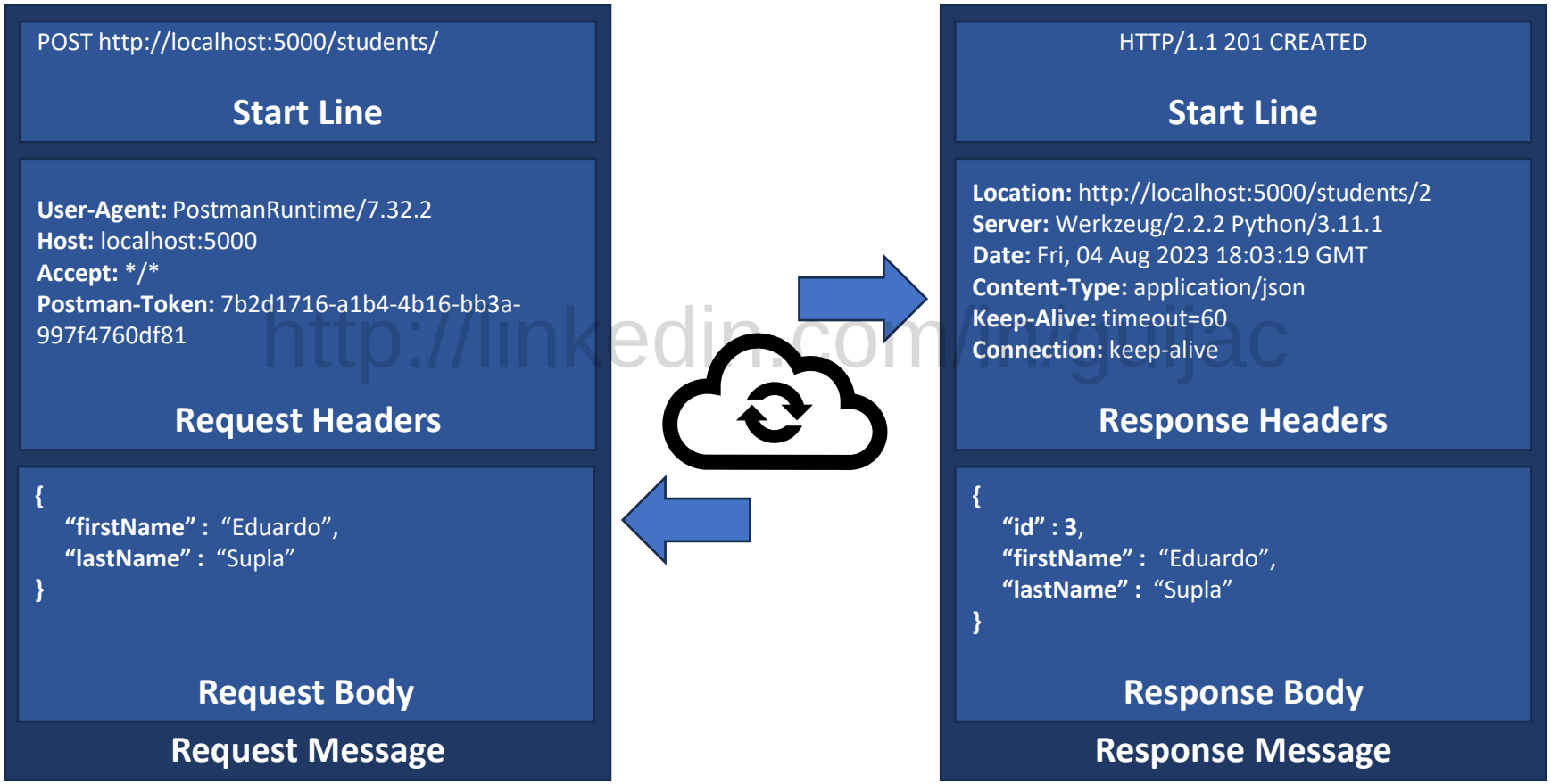
Fluxo simples de troca de mensagens com Protocolo HTTP utilizando verbo/método “GET”

Verbos/Métodos HTTP: POST



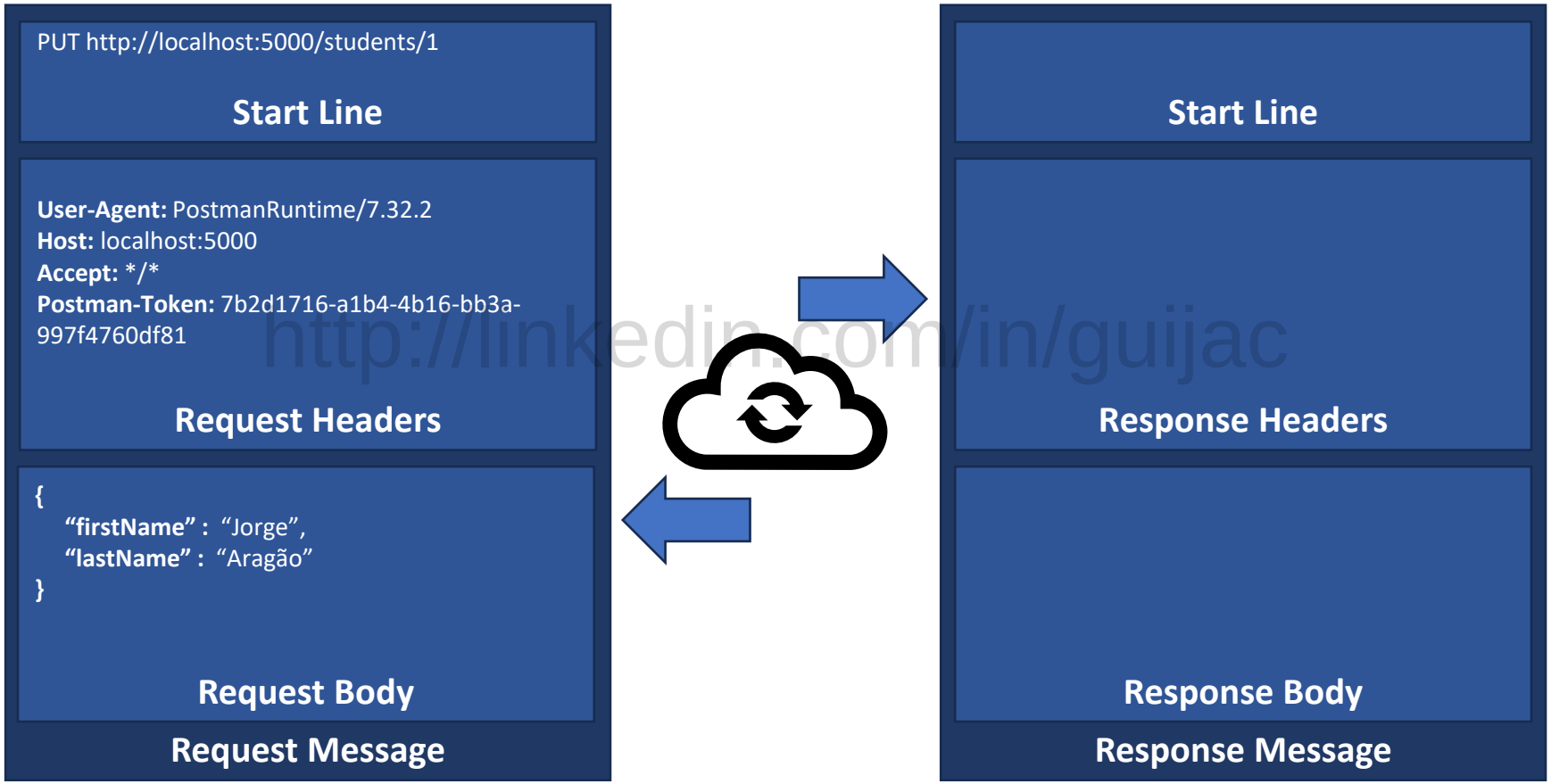
Fluxo simples de troca de mensagens com Protocolo HTTP utilizando verbo/método “POST”

Verbos/Métodos HTTP: POST



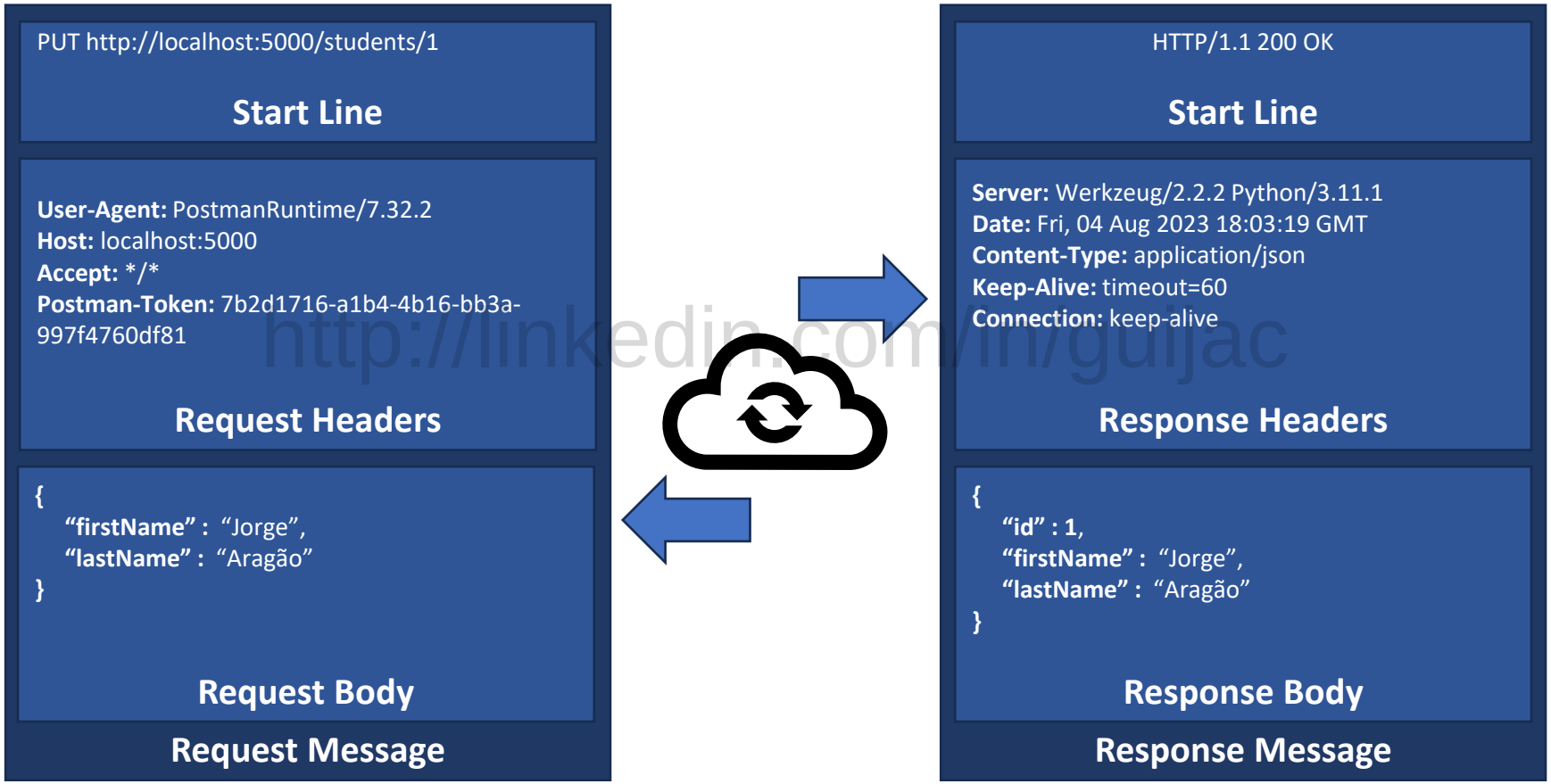
Fluxo simples de troca de mensagens com Protocolo HTTP utilizando verbo/método “POST”

Verbos/Métodos HTTP: PUT



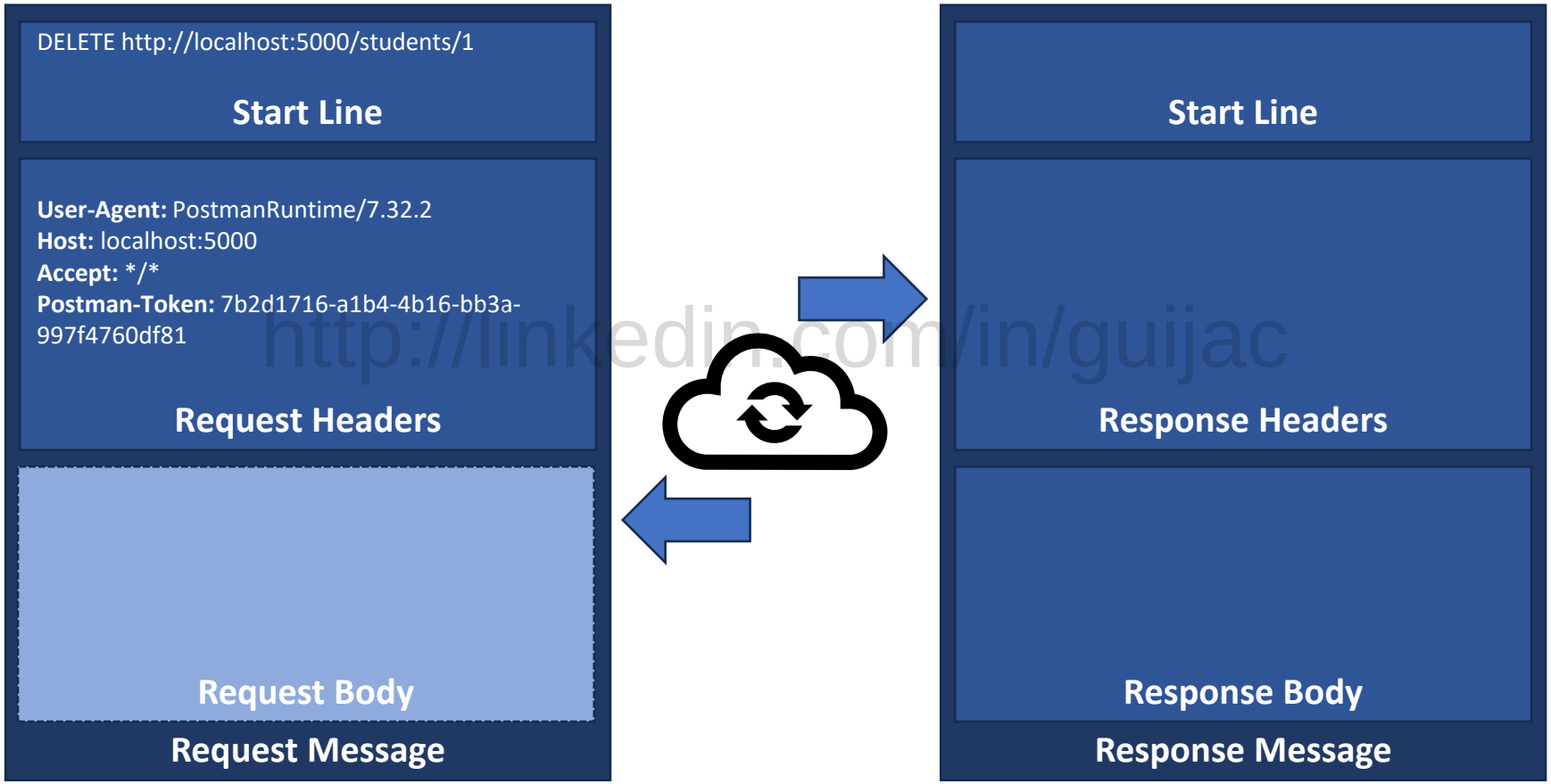
Fluxo simples de troca de mensagens com Protocolo HTTP utilizando verbo/método “PUT”

Verbos/Métodos HTTP: PUT



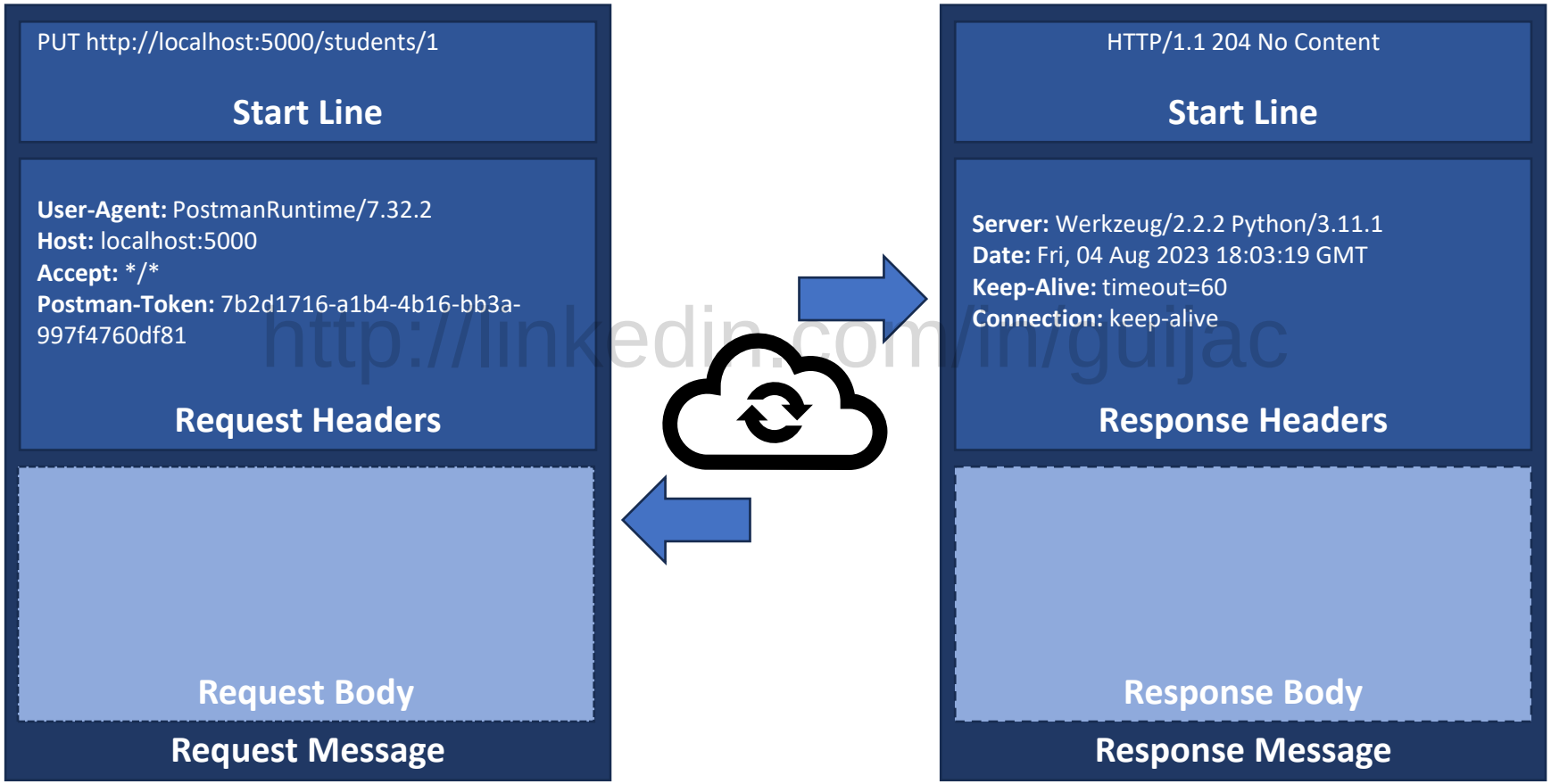
Fluxo simples de troca de mensagens com Protocolo HTTP utilizando verbo/método “PUT”

Verbos/Métodos HTTP: DELETE



Fluxo simples de troca de mensagens com Protocolo HTTP utilizando verbo/método “DELETE”

Verbos/Métodos HTTP: DELETE



Fluxo simples de troca de mensagens com Protocolo HTTP utilizando verbo/método “DELETE”

Verbos/Métodos HTTP: Juntando as Peças

Verbos/Métodos HTTP: Juntando as Peças

Verbos/Métodos HTTP: Juntando as Peças

GET Malone

Verbos/Métodos HTTP: Juntando as Peças

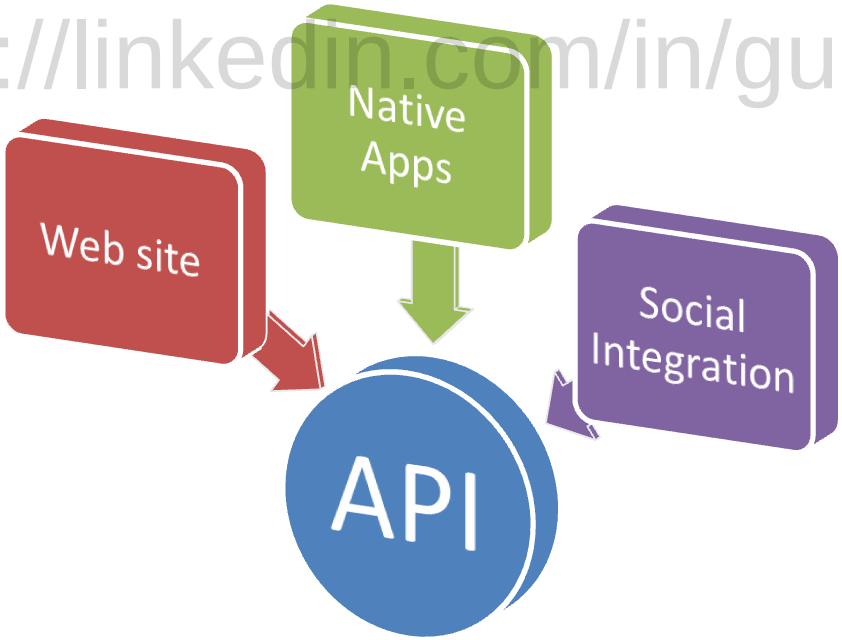
Método	Objetivo	CRUD	Request Body	Idempotente[1]	Seguro[2]
GET	Obter um recurso ou grupo	Read	Não	Sim	Sim
POST	Criar um novo recurso	Create	Sim	Não	Não
PUT	Atualizar um recurso inteiro	Update	Sim	Sim	Não
DELETE	Deletar um recurso	Delete	Não	Sim	Não
PATCH	Atualizar parcialmente um recurso	Update	Sim	Não	Não

GET Malone

[1] Em Matemática ou Ciência da Computação, propriedade de uma operação ser aplicada mais de uma vez sem que haja alteração em seu resultado.
[2] Método que não altera um dado no lado servidor.

Application Programming Interface (APIs)

- ***A**pplication **P**rogramming **I**nterface*. Conjunto de padrões de programação que possibilitam a construção de aplicações, a comunicação entre eles e sua utilização sem a necessidade de possuir conhecimento da maneira como foram implementados (PRESSMAN, 2016).



Fonte: O que é API? REST e RESTful? Conheça as definições e diferenças!

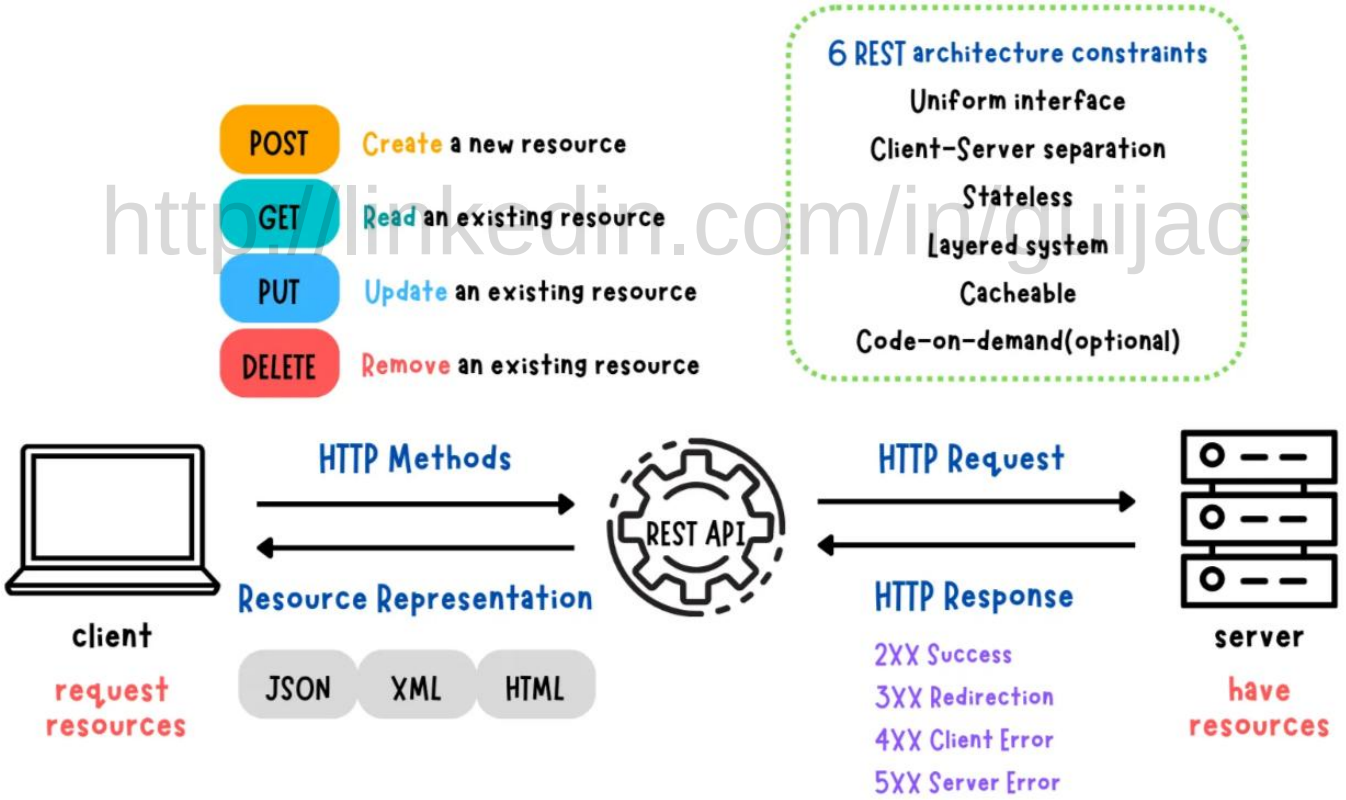
Application Programming Interface (APIs)

- REST (**RE**presentational **S**tate **T**ransfer) é o estilo arquitetural proposto por Roy Thomas Fielding, em **2000**, em sua dissertação de doutorado, baseado em HTTP e composto de algumas restrições (*constraints*) arquiteturais:

  - **Cliente-Servidor**: divide as preocupações de interface com o usuário e armazenamento de dados;
  - **Interface Uniforme**: facilita a evolução independente por meio de mensagens auto descritivas;
  - **Sistema em Camadas**: possibilita melhorias na escalabilidade e segurança;
  - **Armazenamento em Cache**: elimina interações, aumentando escalabilidade e desempenho;
  - **Sem Estado (Stateless)**: exige que o servidor não retenha informações entre requisições, garantindo que cada solicitação contenha todo o necessário.

Application Programming Interface (APIs)

- REST refere-se a um **estilo arquitetural** formalmente definido, RESTful refere-se à **capacidade de um sistema de adotar estes princípios**.



Fonte: Adaptado de REST APIs Explained - 4 Components (mannhowie.com)

O Framework Spring Boot

- Utilizado para desenvolvimento web em Java. Fornece ferramentas modernas para criação de APIs RESTful, com gerenciamento automático de dependências, rotas, e servidores embutidos (como o Tomcat);
- Por ser baseado no **ecossistema Spring**, simplifica a configuração e acelera o desenvolvimento de aplicações Java;

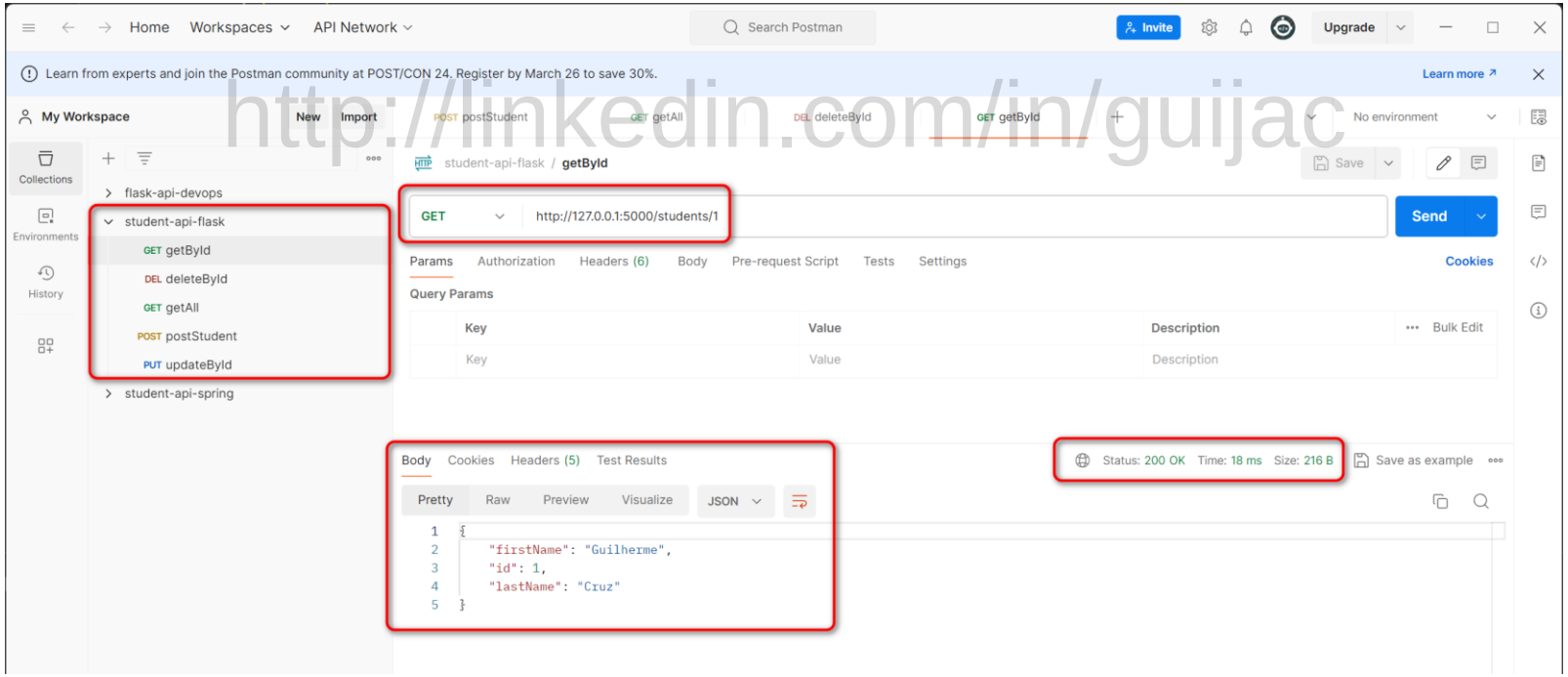
```java
// Importa as anotações do Spring necessárias
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.*;

// Classe principal da aplicação
@SpringBootApplication
@RestController
public class Application {
    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
    // Define uma rota GET acessível pela URL "/"
    @GetMapping("/")
    public String hello() {
        return "Alguma mensagem";
    }
}
```

Exemplo de REST API Java + Spring Boot

Testando REST APIs com Postman

- Aplicação que permite testar REST APIs. Criado em 2012 por Abhinav Asthana, Ankit Sobti e Abhijit Kane em Bangalore, Índia, para resolver o problema de compartilhamento de testes de REST APIs.



Exemplo de uso do Postman testando uma REST API com o verbo/método “GET”. A aplicação organiza-se através de “Collections”, estrutura que contém as informações necessárias para testar uma determinada rota de uma REST API.

Referências Bibliográficas

JSON.org. **Introdução ao JSON**. Disponível em https://www.json.org/json-pt.html. Acesso em 18 mar 2024;

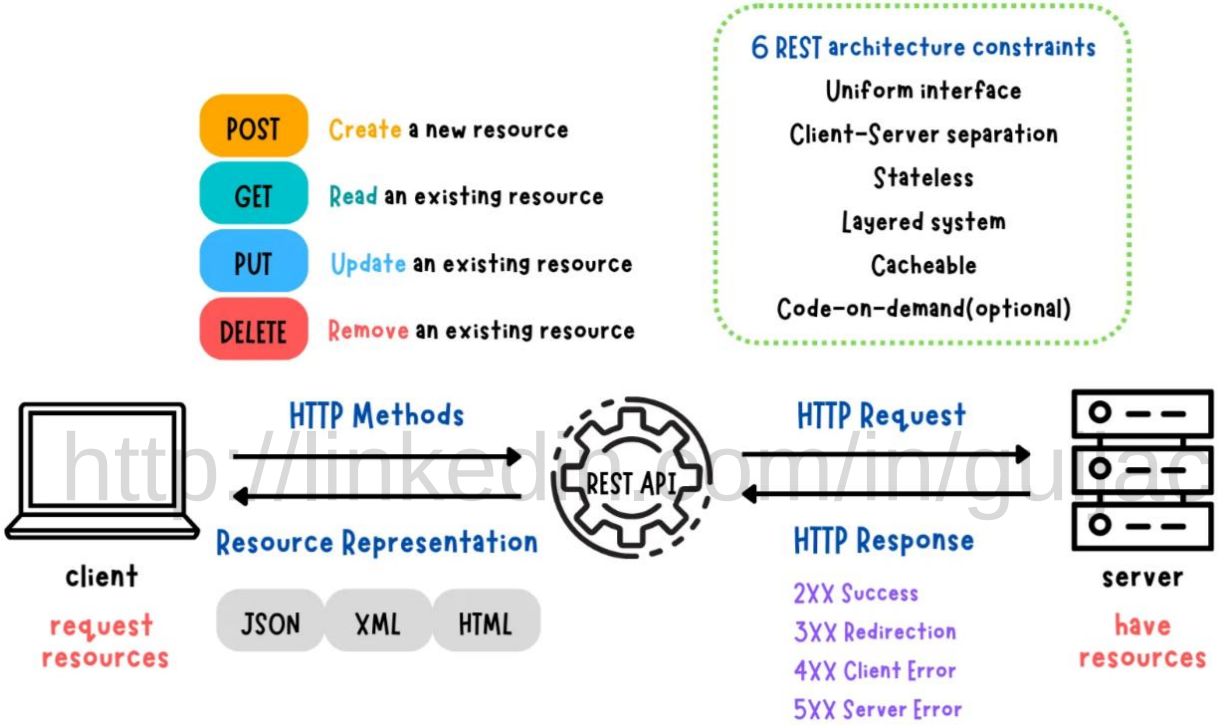
MOZILLA. **HTTP request methods**. Disponível em https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods. Acesso em 18 mar 2024;

FIELDING, Roy T.; TAYLOR, Richard N. **Architectural styles and the design of network-based software architectures**. Irvine: University of California, Irvine, 2000;

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software-8ª Edição**. McGraw Hill Brasil, 2016.

SPRING.io. **Building a RESTful Web Service**. Disponível em https://spring.io/guides/gs/rest-servisse. Acesso em 28 jan 2026

Por hoje (de teoria!) é só!



Fonte: Adaptado de REST APIs Explained - 4 Components (mannhowie.com)

Prof. Esp. Guilherme Jorge Aragão da Cruz

guilherme.cruz@alumni.usp.br

linkedin.com/in/guijac

Licença

Prof. Esp. Guilherme Jorge Aragão da Cruz


guilherme.cruz@alumni.usp.br


linkedin.com/in/guijac