Professores:

Celso Giusti

Daniel Manoel Filho

Marlon Palata Fanger Rodrigues

REST API

SENAI

REST

Rest é um limite arquitetural aplicado em uma API, que define princípios e restrições para a sua construção.

Temos a nomenclatura também conhecida como **RESTful API.**

A diferença entre REST e RESTful é basicamente:

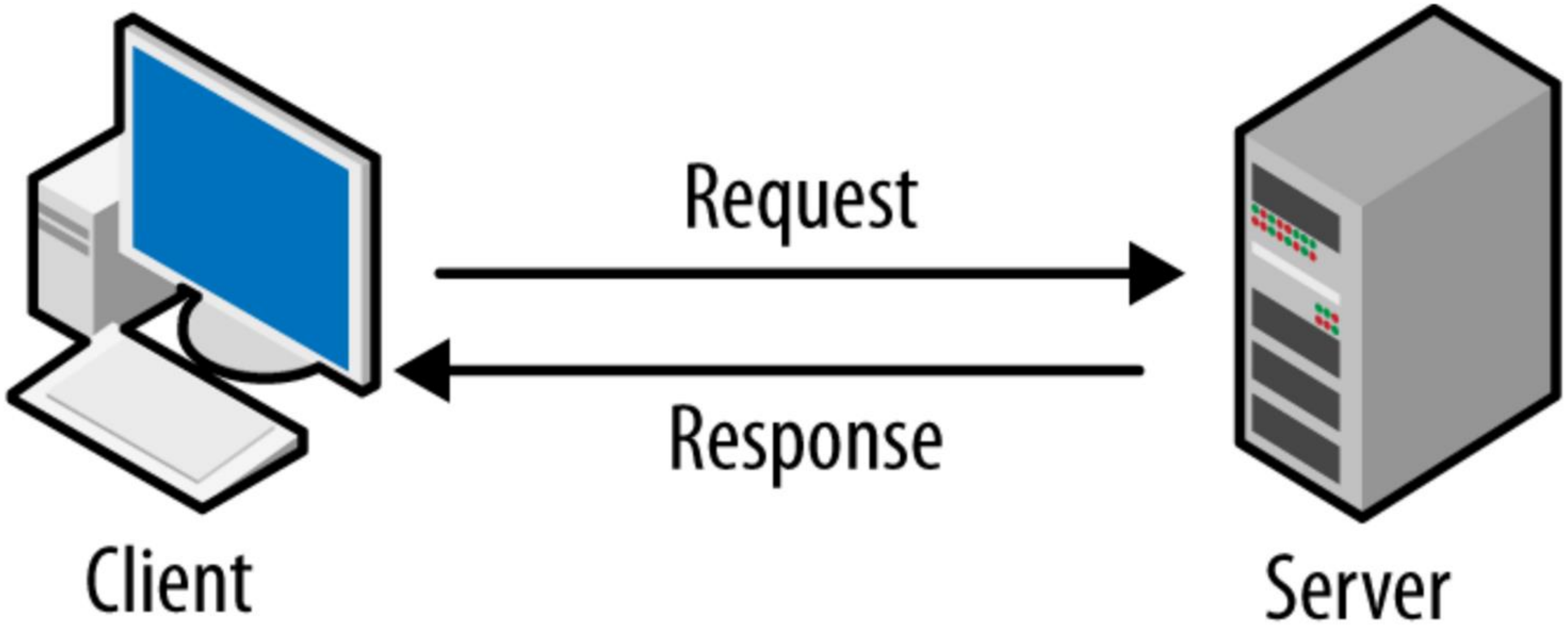
- **REST**: Arquitetura;
- **RESTful**: Uma API construída que segue essa arquitetura.

Vamos conhecer esses princípios:

REST API

Client-Server

O cliente (ex: navegador, app) e o servidor são separados. O cliente faz requisições e o servidor responde. Isso facilita a evolução independente de cada lado e a comunicação.



REST API

Stateless

Cada requisição feita pelo cliente para o servidor deve conter todas as informações necessárias para o servidor entender e responder.

O servidor não guarda as informações da requisição anterior. Isso simplifica o servidor e melhora a escalabilidade.

REST API

Uniform Interface

Essa é a base do REST e define que a comunicação deve ser padronizada usando:

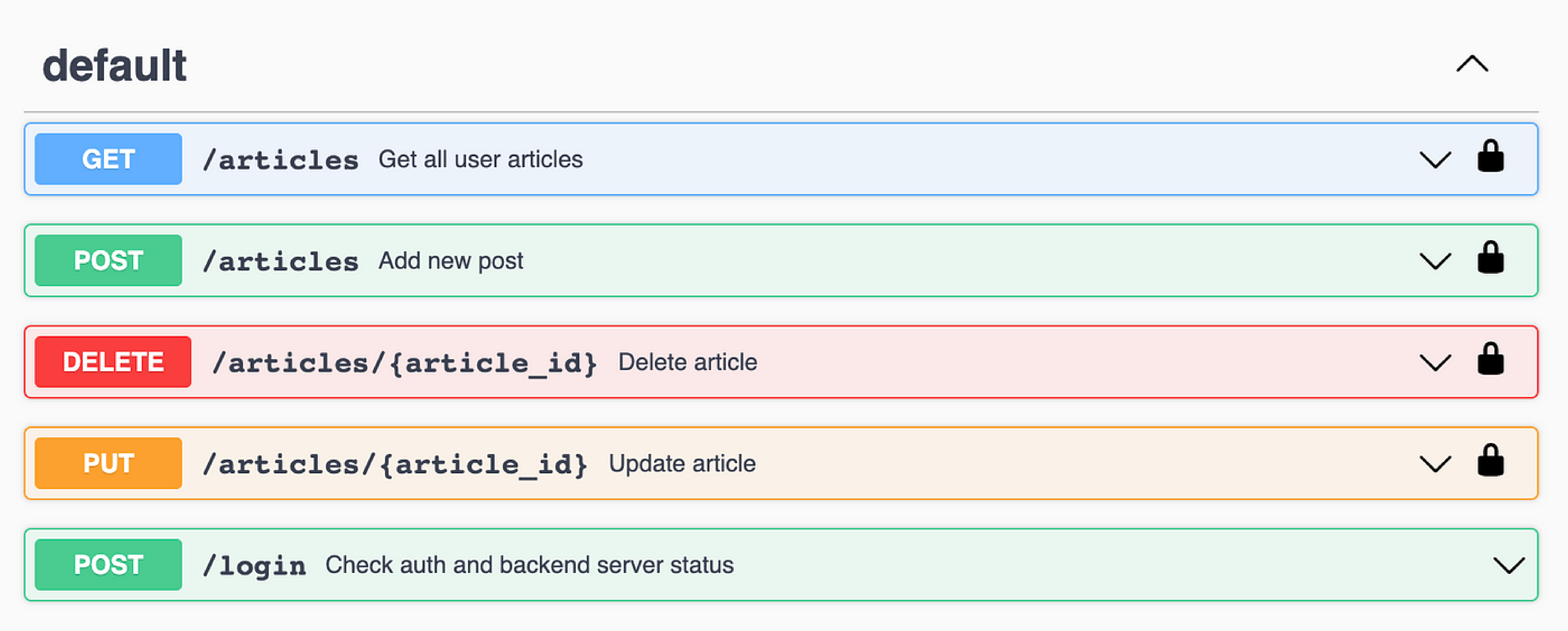
- Recursos identificados por URLs – (ex: localhost/usuarios/123 - identifica o usuário com ID 123);

- Operações padronizadas usando métodos HTTP:
    - **GET**
    - **POST**
    - **PUT**
    - **DELETE**

- Mensagens auto explicativas (requisições e respostas com formatos padronizados, como JSON)

REST API

Métodos HTTP



Componentes

Requisição HTTP

Agora iremos compreender os elementos que compõe uma requisição feita de um cliente para **API**.

Entender esses elementos é uma parte essencial, pois vai esclarecer como podemos consumir e até enviar dados por meio da API e seus métodos.

URL (Uniform Resource Locator)

- Indica o **endereço do recurso**;
- Exemplo: https://api.exemplo.com/

É o **domínio** da API.

Endpoint

O **endpoint** é uma porta de um cômodo específica que iremos acessar para manipular o nosso recurso.

Exemplo: http://dominio.com/usuarios ← Recurso

- Ele indica o que queremos acessar os dados do usuário
- O endpoint representa um **recurso específico na API**.

Componentes

Path Parameter

É um caminho até um item específico.

É como um número de apartamento ou quarto — ele aponta para **um recurso único** dentro do endpoint.

Exemplo: http://dominio.com/usuarios/1 ← ID do usuário

- /usuarios é o recurso;
- /1 é o ID do usuário (um path parameter).

Componentes

Método HTTP

- Indica a **ação** que será feita sobre o recurso;
- Exemplo: **GET**, **POST**, **PUT**, **DELETE**

Pensando que nosso recurso é um usuário, isso define se vamos fazer o cadastro, editar, deletar ou buscar uma lista com esse recurso.

Componentes

Headers (Cabeçalhos)

Enviam informações extras sobre o que está sendo enviado e como deve ser tratado.

- Exemplo:
  - Content-Type: application/json – informa o tipo de dado enviado no body;
  - Authorization: Bearer <token> – Envia tokens de autenticação;
  - Cookie: Envia dados armazenados localmente para o controle de sessão.

No geral o header serve para informar o tipo de conteúdo enviado ou esperado, controlar autenticação e segurança, controlar sessões ou tokens, identificar o cliente (navegador) etc...

Componentes

Body

O conteúdo principal – os dados enviados, como um formulário

- Usado principalmente com POST e PUT;
- Ele acompanha uma estrutura de dados, normalmente em JSON.

JSON – é uma estrutura leve de troca de dados entre sistemas. Muito parecido com objetos em JavaScript.


```json
{
  "id": 1,
  "name": "Desenvolvimento-POSTGRESQL",
  "email": "dev@gmail.com",
  "imageUrl": null
}
```

Componentes

Query Parameters

São filtros ou informações extras colocadas na **URL** para refinar a resposta

- “Quero ver apenas os produtos da categoria **livros**” ou “Só mostre os produtos que possuem estoque > 0” **;**

- Exemplo: GET /produtos?categoria=livros&estoque_maior_que=0a;

Instruções Query Parameters:

- A interrogação representa o início dos query paramaters;
- O formato é representado sempre com chave=valor;
- O “&” representa a separação entre os parâmetros.

Diferença

Path e Query Parameter

Analogia da caixa 📦:

Path Parameter — Qual caixa você quer abrir? Qual o caminho?

Query Parameter — Dentro da caixa, o que você quer ver?

Analogia

Entendendo os componentes

Vamos imaginar que teremos que entregar uma carta, cada componente terá o seu papel:

- URL – é o endereço do prédio para onde você vai enviar a carta;
- Endpoint – é o número da porta ou setor do prédio que vai receber a carta;
- Path Parameter – é o número do destinatário dentro daquele prédio;
- Método HTTP – é o motivo do envio da carta;
- Headers – são etiquetas coladas no envelope, com informações técnicas para o destinatário saber como ler o conteúdo.

Analogia

Entendendo os componentes

- Body – é o conteúdo da carta ou formulário preenchido;

- Query Parameters – São observações escritas fora do envelope, exemplo: “entregar urgente”.

Dessa forma podemos entender de forma mais simples sobre os componentes de uma requisição.

ESCOLA SENAI ÍTALO BOLOGNA

Av. Goiás, 139

Telefone

(11) 2396-1999

Instagram

@senai.itu

Facebok

/senaisp.itu

Site

https://sp.senai.br/unidade/itu/