

O que são Requisições?

Definição:

- Chamadas feitas entre cliente e servidor para troca de dados.

O que é localhost:8080?

Definição:

- é um endereço padrão usado para acessar uma aplicação web rodando localmente (no seu próprio computador)

O que é localhost?

o próprio computador onde a aplicação está rodando. É equivalente ao IP 127.0.0.1.

O que é :8080?

É a porta que está sendo usada para receber conexões. Cada aplicação escuta em uma porta específica. 8080 é uma porta comum para aplicações web, especialmente no Spring Boot.

Exemplo

<http://localhost:8080>

Acessa a raiz da aplicação

<http://localhost:8080/mensagem/enviar>

Acessa a funcionalidade específica
/mensagem/enviar definida no código

E se mudar a porta?

Você pode mudar a porta padrão no arquivo **application.properties** do Spring Boot:

server.port=9090

Métodos mais comuns

- **GET**: Busca informações no servidor (ex: buscar usuários).
- **POST**: Envia dados para o servidor (ex: cadastrar usuário).
- **PUT**: Atualiza dados no servidor (ex: editar usuário).
- **DELETE**: Remove dados do servidor (ex: deletar usuário).

Estrutura básica de uma Response HTTP

- **Status code**: código de status, explica o que aconteceu na requisição
- **Headers**: Parâmetros de cabeçalhos, informações adicionais sobre a resposta.
- **Body**: Corpo da resposta, O body contém os dados reais que o servidor está devolvendo.

códigos de status mais comuns

400	Bad Request	Erro do cliente – dados inválidos ou malformados
404	Not Found	Recurso solicitado não encontrado
200	OK	Requisição bem-sucedida;

O que é CORS?

- CORS (Cross-Origin Resource Sharing) .
- Impede que sites em domínios diferentes façam requisições entre si sem permissão.
- O servidor precisa permitir explicitamente acessos de outros domínios (via headers HTTP)

Uso no conceito java (Spring Boot)

```
package com.exemplo.demo;

import org.springframework.web.bind.annotation.*;

@RestController
@RequestMapping("/api")
@CrossOrigin // habilita CORS para qualquer origem
public class MensagemController {

    @PostMapping("/enviar")
    public String enviarMensagem(@RequestParam String mensagem) {
        return "Mensagem recebida: " + mensagem;
    }
}
```

Uso no conceito java (Spring Boot)

<code>@RestController</code>	Define que essa classe é um controller REST
<code>@RequestMapping("/mensagem")</code>	Define a rota base, ou seja, todas as requisições a <code>/mensagem</code> cairão nessa classe
<code>@PostMapping</code>	Mapeia requisições HTTP do tipo POST
<code>@RequestBody</code>	Lê o corpo da requisição e converte para um objeto
<code>ResponseEntity<String></code>	Permite customizar a resposta HTTP (status, headers, body)