Lidando com CORS em APIs

Cross-Origin Resource Sharing

Rômulo C. Silvestre

September 8, 2025

O Problema: Erros de CORS

A Causa: Same-Origin Policy

É um mecanismo de segurança dos navegadores que, por padrão, **bloqueia** requisições HTTP entre origens diferentes.

O que define uma "Origem"?

Uma origem é a combinação de: **Protocolo + Host + Porta**.

- `https://cursos.edukacode.com.br`
- `http://cursos.edukacode.com.br` → Origem diferente (protocolo)
- `https://faculdade.edukacode.com.br` → Origem diferente (host)
- `http://localhost:3000` vs `http://localhost:8080` → Origem diferente (porta)

A Solução: O Mecanismo CORS

Como Funciona?

CORS permite que o **servidor (API)** informe ao navegador quais origens externas estão autorizadas a acessar seus recursos.

O Cabeçalho HTTP Essencial

A API precisa retornar o header `Access-Control-Allow-Origin` na sua resposta.

- `Access-Control-Allow-Origin: http://localhost:3000`
  - Permite uma origem específica. **(Recomendado)**
- `Access-Control-Allow-Origin: *`
  - Permite qualquer origem. **(Cuidado! Apenas para APIs públicas)**

Habilitando CORS no Spring Boot

A Abordagem Correta

Criar uma classe de configuração global com `@Configuration` que implementa `WebMvcConfigurer`.

Exemplo de Configuração

```
@Configuration
public class CorsConfiguration implements WebMvcConfigurer {

    @Override
    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping("/**") // Aplica a todos os
    endpoints
            .allowedOrigins("http://localhost:3000") //
    Permite o front-end
            .allowedMethods("GET", "POST", "PUT", "DELETE",
                            "OPTIONS", "HEAD", "TRACE", "
    CONNECT");
    }
```