at_S16_A1_SL6_backend

Roteiro de Atividade Prática

Nome: _____ Turma: _____

Título da atividade: Implementação de CORS (cross-origin resource sharing - compartilhamento de recursos entre origens)

1. Pontos principais do conteúdo

CORS (cross-origin resource sharing) é um mecanismo de segurança utilizado por navegadores para restringir solicitações feitas a um servidor que está em um domínio diferente do domínio da página de origem. Por padrão, navegadores bloqueiam essas solicitações por questões de segurança. No entanto, ao configurar corretamente o CORS, é possível permitir que uma API seja acessada de domínios específicos, garantindo maior controle sobre quem pode utilizar seus serviços.

Por que o CORS é importante?
Imagine que você tenha desenvolvido uma API para um sistema de pagamentos e queira permitir que diferentes aplicações possam consumir suas funcionalidades. Sem uma configuração adequada de CORS, essas aplicações podem ser bloqueadas pelo navegador ao tentar acessar a API, mesmo que estejam devidamente autenticadas.

---

2. Explicação detalhada

2.1 Conceitos de CORS

- **Origem**: refere-se ao protocolo (http/https), ao domínio e à porta da aplicação que faz a solicitação.
- **Cabeçalhos de CORS**: utilizados para especificar quais origens têm permissão para acessar a API. Exemplos de cabeçalhos:

- **access-control-allow-origin:** permite que apenas domínios específicos acessem a API;
- **access-control-allow-methods**: define quais métodos HTTP (GET, POST, PUT, DELETE) são permitidos;
- **access-control-allow-headers:** controla quais cabeçalhos personalizados podem ser usados na solicitação.

2.2 Configuração e implementação

- Configurar CORS é essencial em ambientes de produção para evitar que aplicações não autorizadas consumam sua API.
- O CORS pode ser configurado em diferentes servidores, como Apache, Nginx ou através de frameworks como Express.js (JavaScript back-end).

Exemplo prático: imagine que você esteja desenvolvendo uma API para um serviço de análise de crédito. Você deseja permitir que apenas uma aplicação específica hospedada em https://app.exemplo.com possa consumir os dados da API. Configurar o cabeçalho access-control-allow-origin de forma correta garante que outros sites não tenham acesso indevido aos dados dos clientes.

---

3. Roteiro da atividade

Situação-problema: vocês são responsáveis pela implementação de uma API para um sistema de reservas de passagens aéreas. A API precisa ser consumida por diferentes aplicações desenvolvidas por parceiros, mas vocês querem evitar que aplicações não autorizadas acessem o sistema. Configurem as permissões de CORS para garantir que somente parceiros autorizados tenham acesso.

Situações fictícias, produzidas pela SEDUC-SP.

4. Perguntas dissertativas:

1. Qual é a importância do CORS para a segurança de APIs públicas?
2. Quais são os principais cabeçalhos utilizados para configurar o CORS?

3. Como o CORS pode ajudar a controlar o acesso de diferentes origens à sua API?
4. Quais são os riscos de configurar o CORS de forma incorreta?