

Atividade Prática: Demonstrando o CORS (Recusa e Permissão)

Esta atividade te guiará na criação de dois servidores em origens diferentes para demonstrar o problema da Política de Mesma Origem (SOP) e, em seguida, a solução usando o CORS.

Objetivos de Aprendizagem

1. Compreender a **Política de Mesma Origem (SOP)** e o erro "Blocked by CORS Policy".
2. Aplicar o *middleware cors* no Express para **permitir** o acesso de uma origem cruzada específica.
3. Demonstrar a comunicação bem-sucedida entre dois servidores em origens diferentes.

Configuração Inicial

Crie duas pastas separadas: backend e frontend.

1. Pasta backend (API Node.js/Express)

```
mkdir backend
cd backend
npm init -y
npm install express cors
```

2. Pasta frontend (Cliente JavaScript/HTML)

```
mkdir frontend
cd frontend
# O frontend será servido por um servidor simples para garantir uma origem HTTP :
npm init -y
npm install serve
```

Etapa A: O Erro de CORS (A Recusa)

A.1. Backend: backend/server.js (Sem CORS)

Crie o servidor da API que *não* usa o *middleware cors*.



```
1  const express = require('express');
2  const app = express();
3  const PORT = 8080;
4
5  // Rota simples que o frontend tentará acessar
6  app.get('/api/mensagem', (req, res) => {
7    // A API nunca chega aqui se o CORS bloquear a requisição antes.
8    res.json({ data: "Mensagem secreta da API em 8080!" });
9  });
10
11 // Sem middleware CORS configurado!
12
13 app.listen(PORT, () => {
14   console.log(`API Backend rodando em: http://localhost:${PORT}`);
15 })
```

A.2. Frontend: frontend/index.html

Crie o arquivo HTML que tentará consumir a API na porta 8080.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <title>Frontend em 3000</title>
</head>
<body>
  <h1>Cliente (Origem: 3000)</h1>
  <p>Tentando acessar a API (Origem: 8080)...</p>
  <button onclick="fazerRequisicao()">Acessar API</button>
  <pre id="resultado"></pre>

  <script>
    const API_URL = 'http://localhost:8080/api/mensagem';
    const resultadoElement = document.getElementById('resultado');

    async function fazerRequisicao() {
      resultadoElement.textContent = 'Carregando...';
      try {
        // Requisição Fetch para a origem cruzada
        const response = await fetch(API_URL);

        // Se o CORS tivesse permitido, o código continuaria aqui.
        const data = await response.json();
        resultadoElement.textContent = JSON.stringify(data, null, 2);

      } catch (error) {
        // IMPORTANTE: O erro de CORS NÃO é capturado pelo try...catch.
        // O navegador apenas o registra no CONSOLE.
        resultadoElement.textContent = 'Erro de requisição. Cheque o console do navegador!';
        console.error("Erro na requisição (verifique o erro de CORS):", error);
      }
    }
  </script>
</body>
```

A.3. Execução e Verificação (O Erro)

1. Inicie o Backend:

```
# Na pasta backend
node server.js
```

2. Inicie o Frontend na porta 3000 (usaremos o pacote serve):

```
# Na pasta frontend
serve -l 3000
```

3. Acesse <http://localhost:3000> no navegador, abra o **Console** (F12) e clique em **Acessar API**.
4. **Verificação Esperada:** O aluno deve ver no Console do navegador o erro:

"Access to fetch at 'http://localhost:8080/api/mensagem' from origin 'http://localhost:3000' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource."

Etapa B: A Solução de CORS (A Permissão)

B.1. Backend: backend/server.js (Com CORS)

Edite o arquivo server.js para usar e configurar o *middleware* cors, permitindo explicitamente o frontend em http://localhost:3000.

```
const express = require('express');
const cors = require('cors'); // <--- 1. Importa o módulo CORS
const app = express();
const PORT = 8080;

// 2. Configuração do CORS: Permite APENAS a origem do frontend (3000)
const corsOptions = {
  origin: 'http://localhost:3000',
  methods: 'GET',
  optionsSuccessStatus: 200 // Algumas versões legadas precisam disso
};

// 3. Aplica o middleware CORS
app.use(cors(corsOptions));

// Rota simples que o frontend tentará acessar
app.get('/api/mensagem', (req, res) => {
  // AGORA o navegador permite a comunicação, e a API responde
  res.json({ data: "Mensagem secreta da API em 8080 - COM PERMISSÃO CORS!" });
});

app.listen(PORT, () => {
  console.log(`API Backend rodando em: http://localhost:${PORT}`);
  console.log(`AGORA PERMITINDO acesso de: http://localhost:3000`);
})
```

B.2. Execução e Verificação (O Sucesso)

1. Reinicie o Backend:

```
# Na pasta backend, pare (Ctrl+C) e reinicie
node server.js
```

2. **Recarregue o Frontend** (http://localhost:3000) (não precisa reiniciar o serve).
3. Clique em **Acessar API** novamente.
4. **Verificação Esperada:** A requisição deve ser bem-sucedida, e a mensagem: "Mensagem secreta da API em 8080 - COM PERMISSÃO CORS!" deve aparecer no campo de resultado (<pre>).

Perguntas para Reflexão

1. Qual cabeçalho HTTP de **resposta** o *middleware cors* adicionou na Etapa B para que o navegador permitisse a requisição?
2. Se você mudasse o *backend* para a porta 8081 sem mudar o frontend, o CORS ainda bloquearia? Por quê?
3. O que aconteceria se você usasse `origin: '*'` no `corsOptions`?

Entrega da Atividade:

A entrega desta atividade será em duas partes: o código-fonte no GitHub e um vídeo de reflexão demonstrando seu aprendizado.

Prazo: 03/12/2025 até às 21h

1. Submissão do Código-Fonte (GitHub)

Você deve disponibilizar o código completo das pastas *backend* e *frontend* no seu repositório do GitHub. O código deve demonstrar a conclusão da **Etapa B** (solução do CORS) e incluir as modificações realizadas para os desafios opcionais.

- **Instruções:**
 1. Crie um novo repositório (ex: `cors-fundamentals-explorer`).
 2. Faça o *commit* das pastas *backend* e *frontend*.
 3. Inclua um **README.md** detalhando as instruções para rodar o Backend (`node server.js`) e o Frontend (`serve -l 3000`).
 - **Link de Entrega:** Copie a URL do seu repositório principal no GitHub.
-

2. Vídeo de Demonstração e Reflexão (Gravação)

Grave um vídeo (máximo de 3 minutos) onde você responderá às perguntas de reflexão acima e demonstrará a funcionalidade do seu projeto.

Conteúdo do Vídeo:

1. **Apresentação e Demonstração do Sucesso:** Apresente-se e demonstre, no navegador, a comunicação **bem-sucedida** entre seu Frontend (porta 3000) e Backend (porta 8080).
2. **Demonstração do Erro:** Mostre rapidamente o erro de bloqueio da **Política de Mesma Origem (SOP)** na Etapa A (sem CORS) antes de mostrar a correção.
3. **Respostas de Reflexão:** Responda às seguintes perguntas de forma clara e concisa (use o código ou os logs de rede do navegador como apoio visual).