CORS

Autorize a leitura pelo navegador

Pré-requisitos

Conheça cabeçalhos HTTP, fetch e a diferença entre portal e servidor de arquivos. Use Node.js 24 e um navegador. O laboratório tem duas origens locais e retorna JSON fictício para observar a negociação.

Identifique a origem [1]

A origem combina esquema, host e porta. Duas portas no mesmo computador formam origens distintas. A política CORS informa ao navegador se um script de uma origem pode ler uma resposta de outra. Ela complementa a autenticação e a autorização do comprovante, que o serviço precisa aplicar separadamente.

Entenda a consulta preliminar

Um GET com cabeçalho personalizado pode disparar OPTIONS antes da chamada real. O navegador envia a origem, o método pretendido e os nomes dos cabeçalhos solicitados. A resposta indica quais são admitidos. A resposta do GET também precisa indicar uma origem permitida. O status 204 por si só é insuficiente.

Escolha permissões estritas [2]

No S3, configure AllowedOrigins, AllowedMethods e AllowedHeaders conforme o portal. O navegador conduz o preflight e o S3 o avalia com a regra CORS; configure o método real, como GET. Se o portal precisar ler um cabeçalho de resposta que não é exposto por padrão, inclua-o em ExposeHeaders. Evite curingas quando a origem é conhecida.

Modele credenciais e cache

Em uma chamada com credenciais incluídas, use origem explícita e o cabeçalho de credenciais apropriado. Em respostas que dependem da origem, use Vary: Origin para que caches possam distinguir as variantes. Uma URL pré-assinada continua a exigir configuração CORS se o portal ler o corpo com fetch.

CORS

Exercício resolvido • Observe o preflight

Enunciado e dados de partida

O portal está na porta 8788. O serviço de comprovante está na 8789. Permita ao portal fazer GET com X-Agenda-Id. Depois retire o cabeçalho da resposta ao OPTIONS e veja se o browser chega ao GET.

1. Salve cors.mjs e rode node cors.mjs

```
import { createServer } from 'node:http';

const portal = 'http://127.0.0.1:8788';
createServer((req, res) => {
  res.setHeader('Content-Type', 'text/html; charset=utf-8');
  res.end('<h1>Portal da agenda</h1>');
}).listen(8788, '127.0.0.1');

createServer((req, res) => {
  res.setHeader('Vary', 'Origin');
  if (req.headers.origin === portal) {
    res.setHeader('Access-Control-Allow-Origin', portal);
  }
  if (req.method === 'OPTIONS') {
    res.setHeader('Access-Control-Allow-Methods', 'GET');
    res.setHeader('Access-Control-Allow-Headers', 'X-Agenda-Id');
    res.statusCode = 204;
    return res.end();
  }
  res.setHeader('Content-Type', 'application/json');
  res.end(JSON.stringify({ reserva: 'R45', estado: 'disponivel' }));
}).listen(8789, '127.0.0.1');
```

Leia o servidor

O cabeçalho de origem é emitido nas duas respostas quando a origem permitida coincide. O OPTIONS inclui o método e o cabeçalho. O exemplo não implementa credenciais nem entrega PDFs reais.

CORS

Confira a negociação e o diagnóstico

2. Abra http://127.0.0.1:8788 e use o console

```
const resposta = await fetch('http://127.0.0.1:8789/comprovante', {
  headers: { 'X-Agenda-Id': 'R45' },
});
console.log(await resposta.json());
```

Leia os resultados

Na aba Rede, observe OPTIONS com 204 e GET com 200. No console, o objeto tem reserva R45 e estado disponivel. O exemplo não configura um prazo para o cache de preflight. Faça variações em uma nova sessão do navegador para evitar reaproveitar uma decisão anterior.

Caso de erro • Falta permissão para o cabeçalho

Apague a linha Access-Control-Allow-Headers, reinicie o servidor e repita. O preflight recebe 204, mas fetch rejeita a leitura e o navegador não realiza o GET dessa tentativa. Restaure a regra. Se o erro estiver na resposta do GET, ele pode chegar ao servidor e mesmo assim o navegador impedir que o script leia a resposta.

Variação e critérios

Troque o valor de portal no servidor para uma origem diferente e mantenha o navegador na porta 8788. A resposta fica sem Access-Control-Allow-Origin para essa origem e o preflight falha. Confira método, cabeçalho e origem no navegador. Uma chamada direta com curl pode ajudar a inspecionar o HTTP, mas a decisão CORS do navegador exige teste no navegador. Encerre o servidor com Ctrl+C.

Referências primárias

[1] WHATWG • Fetch Standard, protocolo CORS

[2] AWS • Elementos da configuração CORS no S3

CORS

Diagnostique o preflight no navegador

Confira origem, método e cabeçalhos

Um PUT de upload pode disparar OPTIONS antes do envio. A regra CORS precisa aceitar a origem da interface, o método e os cabeçalhos solicitados. A autorização da operação continua dependendo da assinatura e das permissões do bucket.

Exercício adicional

A URL funciona com um cliente de terminal, mas falha no navegador da homologação. O que comparar?

Resolução

Abra a aba de rede e confira Origin e os cabeçalhos do preflight. Ajuste a regra para a origem autorizada e somente os métodos e cabeçalhos usados. Repita o teste pelo navegador.

Ao assistir ao vídeo

Identifique quais decisões pertencem ao navegador e quais cabeçalhos vêm do servidor.

Vídeo complementar

Web Dev Simplified - Learn CORS In 6 Minutes (em inglês)