



El problema de CORS 11/23



RECURSOS

MARCADORES

Cross Origin Resource Sharing (CORS) o en español, **Intercambio de Recursos de Origen Cruzado**, es un mecanismo de seguridad para la solicitud e intercambio de recursos entre aplicaciones con dominios diferentes. En pocas palabras, si las solicitudes HTTP se realizan desde un dominio diferente al dominio del backend, estas serán rechazadas.

Si eres desarrollador o desarrolladora front-end, tendrás problemas de **CORS** a lo largo de tu carrera y en múltiples oportunidades. Pero no te preocupes, es completamente normal y vamos a ver de qué se trata para evitar dolores de cabeza.

Cómo habilitar el dominio

Si **CORS** no está habilitado en el backend que estés consultando, las peticiones se bloquearán y verán un error en la consola de desarrollo del navegador.

Dependerá del equipo back end o de ti si también estás desarrollándolo, de habilitar el dominio del front-end desde el cual se ejecutarán las peticiones.

La habilitación del dominio suele ser muy sencilla y dependerá del lenguaje de programación y *framework* que estés utilizando, pero suele verse de la siguiente manera:

```
// MAL: Un * da permisos a cualquier dominio de realizar peticiones, es una muy mala
práctica de seguridad.
cors({ origin: '*' })










// BIEN: Lo ideal es solo permitir los dominios que queremos autorizar a realizar
peticiones.
cors({ origin: ['mydomain.com', 'app.mydomain.com'] })
```


Entornos donde se produce este error

Este error suele producirse principalmente en entornos de desarrollo o productivos reales, en servidores en la nube. No es tan habitual que suceda en entornos locales, ya que aplicaciones como Postman o Insomnia para realizar pruebas de petición HTTP, cambian el origen de las peticiones y evitan este problema.

Contribución creada por: Kevin Fiorentino.

Archivos de la clase

 /	
 tsconfig.app.json	
 package.json	
 angular.json	
 .eslintrc.json	

 package-lock.json




 .browserslistrc



 .gitignore



 obs-promises.js




 tsconfig.spec.json



 proxy.config.json



 tsconfig.json




 README.md



 .editorconfig



 karma.conf.js



 src

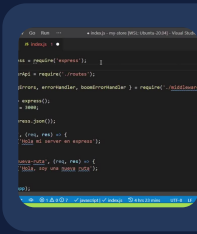
Lecturas recomendadas



Cross-Origin Resource Sharing (CORS) - HTTP | MDN
<https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>



Clases relacionadas




```
const express = require('express');
const cors = require('cors');
const app = express();
const port = 3000;

app.use(cors());
app.use(express.json());

app.get('/', (req, res) => {
  res.json({ message: 'Hello World' });
});

app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

Problema de CORS

 Curso de Backend con Node.js: API REST con Express.js