# Access-Control-Allow-Origin: Dealing with CORS Errors in React and Express

<https://daveceddia.com/access-control-allow-origin-cors-errors-in-react-express/>

Getting this error in your React and/or Express app?

*No ‘Access-Control-Allow-Origin’ header is present on the requested resource.*

You’ve run afoul of the **Same Origin Policy** – it says that every AJAX request must match the exact host, protocol, and port of your site. Things that might cause this:

* Hitting a server from a locally-served file (a request from file:///YourApp/index.html to [http://api.awesome.com](http://api.awesome.com/))
* Hitting an external API (a request from [http://yourapp.com](http://yourapp.com/) to [http://api.awesome.com](http://api.awesome.com/)).
* Hitting an internal API (a request from [http://yourapp.com](http://yourapp.com/) to [http://api.yourapp.com](http://api.yourapp.com/)).
* Hitting a different port on the same host (webapp is on [http://localhost:3000](http://localhost:3000/), API is [http://localhost:4000](http://localhost:4000/))
* Requesting over http from https or vice-versa (requesting [https://yourapp.com](https://yourapp.com/) from [http://yourapp.com](http://yourapp.com/))

To be clear, this is not a React error. It afflicts all web apps equally, and most of the fixes we’ll look at below are actually modifying the server or the browser.

## How to fix it

Here are a few ways to solve this problem:

### Best: CORS header (requires server changes)

CORS (Cross-Origin Resource Sharing) is a way for the server to say “I will accept your request, even though you came from a different origin.” This requires cooperation from the server – so if you can’t modify the server (e.g. if you’re using an external API), this approach won’t work.

Modify the server to add the header Access-Control-Allow-Origin: \* to enable cross-origin requests from anywhere (or specify a domain instead of \*). This should solve your problem.

#### How to Enable CORS on Express

If you’re using Express, the easiest way to enable CORS is with the [cors](https://daveceddia.com/access-control-allow-origin-cors-errors-in-react-express/var%20cors%20=%20require('cors')) library.

You just need to install it in your Express project with npm install cors, then require it and add it as a middleware:

var express = require('express');

// Import the library:

var cors = require('cors');

var app = express();

// Then use it before your routes are set up:

app.use(cors());

Two important things to know here:

* Express runs its middleware in order. So make sure this app.use code runs before you set up your routes.
* By default, the cors library will allow requests from any origin. This can open you up to security problems and abuse.

For production use, it’s best not to allow all origins. Instead, create a whitelist of allowed domains, and check each request against the whitelist. Here’s how:

// Set up a whitelist and check against it:

var whitelist = ['[http://example1.com](http://example1.com/)', '[http://example2.com](http://example2.com/)']

var corsOptions = {

origin: function (origin, callback) {

if (whitelist.indexOf(origin) !== -1) {

callback(null, true)

} else {

callback(new Error('Not allowed by CORS'))

}

}

}

// Then pass them to cors:

app.use(cors(corsOptions));