Contents

[How to deal with CORS error in express Node.js Project ? 1](#_Toc106824795)

[*What is Cors:* 1](#_Toc106824796)

[**How To Enable CORS?** 1](#_Toc106824797)

[*Code:* 2](#_Toc106824798)

[Output: 3](#_Toc106824799)

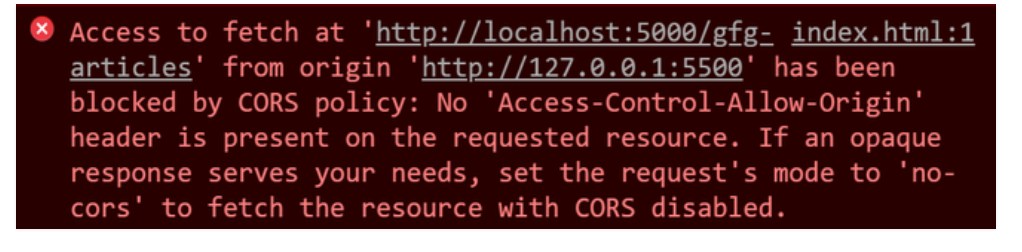
# How to deal with CORS error in express Node.js Project ?

# *What is Cors:*

*CORS, also known as Cross-Origin Resource Sharing, should be enabled if you want to make a request between your client and server when they are on different URLs.*

*Let us consider client to be on****http://localhost:5500****and the server on****http://localhost:5000****. Now if you try to make a request from your client to the server you will get the error stating that blocked by the CORS policy.*

***ERROR:***



## **How To Enable CORS?**

We will use cors, a node.js package to enable CORS in express Node.js Project.

Install the dependency modules using the following command.

npm i express cors

### *Code:*

const express = require('express')// Express module just as you require other modules and puts it in a variable. var app = express(); => Calls the express function "express()" and puts new Express application inside the app variable (to start a new Express application)

const app = express();//create instance of express

const port = 3000 // server will run on this port (You can change the port number )

//NB: same port number you can't run different node project at the same time.

const cors = require('cors');

// CORS is enabled for the selected origins

let corsOptions = {

    origin: [ 'http://localhost:5500', 'http://localhost:3000' ]

};

// If you just want to allow a particular origin to access your site, then corsOptions will be as follows:

// let corsOptions = {

//     origin: 'http://localhost:5500'

//};

// Using cors as a middleware

app.get('/gfg-articles',cors(corsOptions),

    (req,res) => res.json('gfg-articles'))

//Server setup

app.listen(port, () => {

  console.log(`Example app listening on port ${port}`)

})

### Output:

