**Axios-demo @ 28 June 2019**

Creat folder axios-demo

Create sub folder server

Cd server

Npm init -y

Yarn add express cors body-parser

# **Working on the server**

Create index.js

Add to index.js

const express = require('express')

const bodyParser = require('body-parser')

const cors = require('cors')

initialise the app and the port number

const app = express()

const port = 4000

Set up the app to run and listen

app.listen(port, () => {

console.log(`server is listenign on ${port}`)

})

Define students array

const students = []

set up routes

// Routes - set up a simple API

app.get('/api/students', (req, res) => {

return res.json(students)

})

// add a student

app.post('/api/students', (req, res) => {

students.push(req.body.student) // add a student from the incoming request object

return res.json(req.body.student) // print out the

})

Add middleware

// Middleware

app.use(bodyParser.json())

Test with postman to make sure that the routes are working and the api server is operational. Now you can start developing on React knowing that any remaining issues will be with the React-side.

# **Working on the Client-side**

create-react-app client

yarn add axios

in app.js, import axios

import axios from "axios"

in app.js, initialise a constant to set up a connection the API server

const studentAPI = "http://localhostL4000/api/students"

strip out all of the app function in app.js

Add a react class component called app in app.js

export default class App extends React.Component {

  render() {

    return <div />

  }

}

With a class component, start with a constructor or set up the state first.

Set up the state in app.js

  state = {

    students: []

  }

Render each student in

    return (

      <div>

        <h1>Students</h1>

        {this.state.students.map((student, index) => (

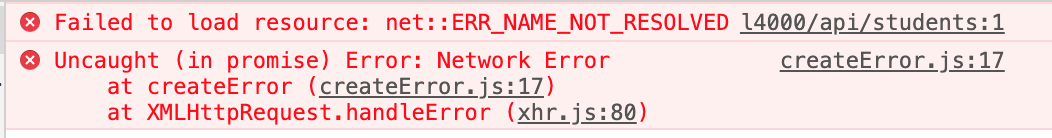
          <h4>{student}</h4>

        ))}

      </div>

    )

This is why we need CORS. Even if they are both on localhost, they are on different ports so they are on different origins.



Go back to the server, go to app.js, add to middleware section

app.use("/api/", cors())

In a larger application, you want to modularise the API.

Create in react client, studentAPI.js

// axios to the project

import axios from "axios"

// axios let you create more than one instance

export default axios.create({

baseURL: "http://localhost:4000/api/students"

// add any objects to configure the axios

})

Remove those from the api.js

Import the studentAPI into app.js

import studentAPI from "./studentAPI"

in App.js, change all instances of axios with studentAPI

  componentDidMount = () => {

    studentAPI.get().then(result => {

      this.setState({

        // the result will have a .data property

        // this overrides the existing student array with the new data

        students: result.data

      })

    })

  }

in the get and post method in App.js, remove passing in the studentAPI (original const address) because that is no longer required

    studentAPI

      .post("", {

        student: this.state.studentName // axios will automatically stringify the object

      })

Put – tag on an id

studentAPI.post(“/id”, { student: this.state.studentName})

Still need to pass an empty string in the post method