Problem one:

Setting the path correctly: see email

Problem 2: getSingleState to work.

getSingleState: Searching the JSON info and returning single states.

Returns error on browser that says, “Parse is not defined”.

Solution: in assigning the state const, I had put in parse() because the example of employees required a parseInt. It wasn’t needed in this case. After much trial and error, removing it solved the problem.

const getSingleState = (req, res) => {

    const state = data.states.find(cd => cd.code === parse((req.params.state)));

to this:

const getSingleState = (req, res) => {

    const state = data.states.find(cd => cd.code === (req.params.state));

Problem 3:

Trying to make StateCapital function I forgot to pass in the (req, res) parameters when I was calling getSingleState inside it. I called it like this:

getSingleState() instead of getSingleState(req, res)

That’s an hour and a half wasted lol.

Problem 4:

Connecting to the database

Little errors like this. It didn’t work because I originally had this

const dotenv = require('dotenv').config;

instead of this.

const dotenv = require('dotenv').config();

I get confused often about what things actually are. Is config a property? A function? Its not obvious to me which is why it its mysterious and difficult to make things work sometimes.

Updating the database: how to pull up indexes?

const result = await funfact.updateOne({code: req.params.state && funfact[req.body.index]} , {funfact: req.body.funfact},{upsert:false});

I switched it up and tried the way dave shows on the video. I still couldn’t get it. I finally figured it out with a handy little node.js function isArray(), which revealed why it wouldn’t work. I was passing an index to something that isn’t an array! Its an object! A little more expirementation and boom:

 const index = req.body.index;

    funFactUpdate.funfact[index] = req.body.funfact;

Trying to merge DB with static json:

Had major problems. After figuring out what the spread operator is, I still couldn’t get it. I was having a very strange problem, where I could console.log the correct info to the screen but I couldn’t send it as json to the browser. After a minute I figured out it was because of async await problems. The function was sending the json, unmerged first, then going back and merging the data for the console.log

Turns out it has something to do with Async Await. I figured that out because the console.log and the return values were different. After messing around for a few hours I gave up and went back to review Async await. I youtubed it, and Dave’s video popped up!!

forEach() doesn’t return anything, so it can’t be used with async functions!