Lab 5

Due Feb 23, 2022 by 11:59pm **Points** 100 **Submitting** a file upload

CS-546 Lab 5

JSON Routes

For this lab, you will create a simple server that will provide data from an API.

For this lab, you will not need to use a database.

For this lab, you **must** use the `async/await` keywords (not Promises). You will also be using `axios` (https://github.com/axios/axios) , which is a HTTP client for Node.js; you can install it with `npm i axios`. You will use it just as you did in lab 3.

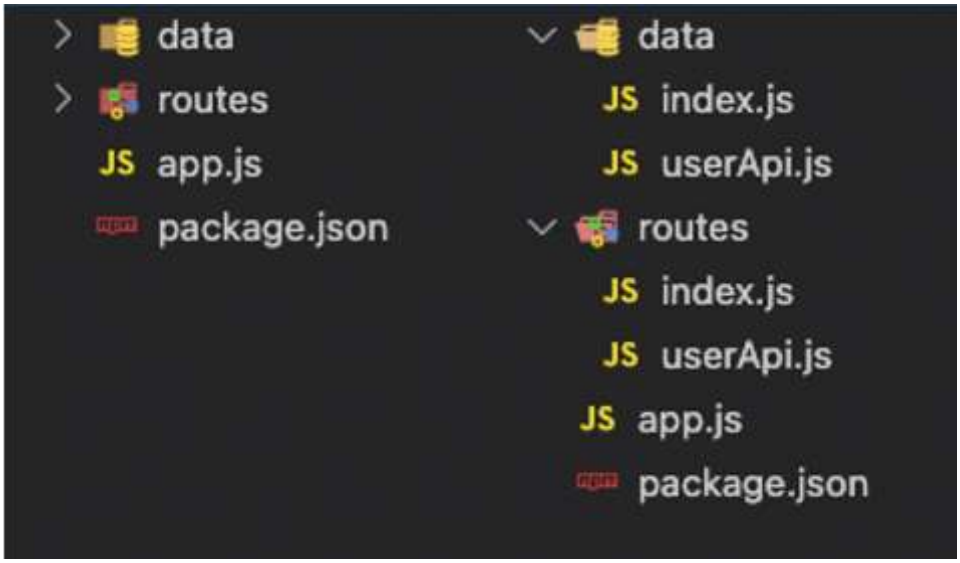
Network JSON Data

You will be downloading JSON files from the following GitHub Gists:

- people.json (https://gist.githubusercontent.com/graffixnyc/31e9ef8b7d7caa742f56dc5f5649a57f/raw/43356c676c2c
- work.json (Links to an external site.) (https://gist.githubusercontent.com/graffixnyc/febcdd2ca91ddc685c163158ee126b4f/raw/c9494f59261

Folder Structure

You will use the following folder structure for this lab:



In Lecture Code for Lab 5, we worked with MongoDB. For this lab, you will get data from Axios calls and should modify the data folder accordingly.

Your routes

`/people`

When making a GET request to `http://localhost:3000/people`, this route will return the JSON data that is returned from the axios call to the URL endpoint. You will use `people.json` for the list of people. You MUST return the data in JSON format.

`/work`

When making a GET request to `http://localhost:3000/work`, this route will return the JSON data that is returned from the axios call to the URL endpoint. You will use `work.json` for the list of companies. You MUST return the data in JSON format.

`/people/:id`

When making a GET request to `http://localhost:3000/people/:id`, this route will return the JSON data. You will use people.json Where `:id` is the parameter that is passed to the route: http://localhost:3000/people/479 This endpoint returns a JSON object that has all the details for the person with that with the supplied `:id` **If the ID cannot be found in the Data(i.e. there is no person with that ID), or if the URL parameter is any other data type besides a positive whole number (all URL params are strings, so you will need to try to convert the URL param to a number), you will throw an error. You MUST return the data in JSON format.**

`/work/:id`

When making a GET request to `http://localhost:3000/work/:id`, this route will return the JSON data that is returned from the axios call to the URL endpoint. You will use `work.json` Where `:id` is the parameter that is passed to the route: http://localhost:3000/work/729 This endpoint returns a JSON object that has all the details for the company with that with the supplied `:id` **If the ID cannot be found in the Data(i.e. there is no company with that ID), or if the URL parameter is any other data type besides a positive whole number (all URL params are strings, so you will need to try to convert the URL param to a number), you will throw an error. You MUST return the data in JSON format.**

Packages you will use:

You will use the **express** package as your server.

You will use the **axios** package to get data from the API.

You can read up on **express (http://expressjs.com/)** on its home page. Specifically, you may find the **API Guide section on requests (http://expressjs.com/en/4x/api.html#req)** useful.

You may use the **lecture 5 code (https://github.com/stevens-cs546-cs554/CS-546/tree/master/lecture_05/code)** as a guide.

You must save all dependencies to your package.json file

Requirements

1. You **must not submit** your node_modules folder
2. You **must remember** to save your dependencies to your package.json folder
3. You **must remember** to update your package.json file to set `app.js` as your starting script!
4. You **must** submit a zip archive or you will lose points, named in the following format: `LastName_FirstName_CS546_SECTION.zip`. You will lose points for not submitting an archive named this way