Lab 5

Due Oct 14, 2021 by 11:59pm **Points** 100 **Submitting** a file upload
Available after Oct 7, 2021 at 12am

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/a1196cbf008e85a8e808dc60d4db7261/raw/9fd0d1a4d7846b
- stocks.json (Links to an external site.) (https://gist.githubusercontent.com/graffixnyc/8c363d85e61863ac044097c0d199dbcc/raw/7d79752a9342a

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.

`/stocks`

When making a GET request to `http://localhost:3000/stocks` , this route will return the JSON data that is returned from the axios call to the URL endpoint. You will use `stocks.json` for the list of stocks. 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/4c57ua2a-5f3d-4309-b81c-2f6b36965ecc 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 valid string, you will throw an error. You MUST return the data in JSON format.**

`/stocks/:id`

When making a GET request to `http://localhost:3000/stocks/:id`, this route will return the JSON data that is returned from the axios call to the URL endpoint. You will use `stocks.json` Where `:id` is the parameter that is passed to the route: http://localhost:3000/stocks/929686a2-dd3a-42c7-a88d-b170e2590252 This endpoint returns a JSON object that has all the details for the stock with that with the supplied `:id` **If the ID cannot be found in the Data(i.e. there is no stock with that ID), or if the URL parameter is any other data type besides a valid string, 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.