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`, 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 except the URL you pass to axios will be different than lab 3's

TV Maze API

For this lab, you will be using two endpoints of the TV Maze API which is an API about TV shows for your Axios calls. The list of shows: http://api.tvmaze.com/shows and then you'll get an individual show using the endpoint http://api.tvmaze.com/shows/:id where `:id` is the ID of the show you are looking up.

You will use these two endpoints to make your axios.get calls depending on which route is called.

Your routes

`/shows`

When making a GET request to `http://localhost:3000/shows`, this route will return the JSON data that is returned from the axios call to the URL endpoint. The url you will use for this route that axios will get the data from is http://api.tvmaze.com/shows This endpoint returns list of shows which is an array of objects. Your route will simply return all the `{ data }` that axios returns for the endpoint URL.

`/shows/:id`

When making a GET request to `http://localhost:3000/shows/:id`, this route will return the JSON data that is returned from the axios call to the URL endpoint. The url you will use for this route that axios will get the data from is http://api.tvmaze.com/shows/:id Where `:id` is the parameter that is passed to the route: http://localhost:3000/shows/1 for example would query the endpoint http://api.tvmaze.com/shows/1 This endpoint returns an object that has all the details for a show with that ID: **If the ID cannot be found in the TV Maze API (i.e. there is no show with that ID), or if the URL parameter is any other data type besides a positive whole number, you will throw an error.**

`/aboutme`

When making a GET request to `http://localhost:3000/aboutme`, this route will return JSON in the following structure (with your own information): You can store this object directly in your route and then return it.

```
{
  "name": "Your Name",
  "cwid": "Your CWID",
  "biography": "2 biography paragraphs separated by a new line character (\n).",
  "favoriteShows": ["array", "of", "favorite", "shows"]
}
```

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 on its home page. Specifically, you may find the API Guide section on requests useful.

You may use the lecture 5 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, rar, tar.gz, or .7z archive or you will lose points, named in the following format: `LastName_FirstName_CS546_SECTION.zip` (or, whatever the file extension may be). You will lose points for not submitting an archive.

◂ Previous

Next ▸