Lab 5

Start Assignment

Due Thursday by 11:59pm **Points** 100 **Submitting** a file upload **File Types** zip

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

Pokémon API

For this lab, you will be using two endpoints of the Pokémon API. for your Axios calls. The list of Pokémon: https://pokeapi.co/api/v2/pokemon (https://pokeapi.co/api/v2/pokemon) (http://api.tvmaze.com/shows) and then you'll get an individual show using the endpoint https://pokeapi.co/api/v2/pokemon (https://pokeapi.co/api/v2/pokemon)/:id where `:id` is the ID of the Pokémon you are looking up.

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

It is important that you look at the API documentation and understand the the schema of the data being returned: https://pokeapi.co/docs/v2 (https://pokeapi.co/docs/v2) **. When working with an API it's ALWAYS super important that you understand the data you are working with. The schema, the format of the data etc.. READ THE DOCUMENTATION.**

Folder Structure

You will use the folder structure in the stub for the data & routes module, and other project files. There is an extra file in the stub called helpers.js. You can add all your helper/validation functions in that file to use in your other modules.

YOU MUST use the directory and file structure in the code stub, or points will be deducted. You can download the starter template here: lab5_stub.zip (https://sit.instructure.com/courses/61549/files/10351550/download?download_frd=1) **PLEASE NOTE: THE STUB DOES NOT INCLUDE THE PACKAGE.JSON FILE. YOU WILL NEED TO CREATE IT! DO NOT ADD ANY OTHER FILE OR FOLDER APART FROM PACKAGE.JSON FILE.**

Your routes

`/pokemon`

When making a GET request to `http://localhost:3000/pokemon`, 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 https://pokeapi.co/api/v2/pokemon (https://pokeapi.co/api/v2/pokemon) (http://api.tvmaze.com/shows). This endpoint returns list of Pokémon. Your route will simply return all the `{ data }` that axios returns for the endpoint URL. The https://pokeapi.co/api/v2/pokemon (https://pokeapi.co/api/v2/pokemon) route only returns the first 20 Pokémon in the API but there are a lot more than 20 Pokémon in the API. This route only needs to return the initial list of the first 20 Pokémon in the API.

`/pokemon/:id`

When making a GET request to `http://localhost:3000/pokemon/: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 https://pokeapi.co/api/v2/pokemon/:id (https://pokeapi.co/api/v2/pokemon) Where `:id` is the parameter that is passed to the route: http://localhost:3000/pokemon/1 (http://localhost:3000/pokemon/1) for example would query the endpoint https://pokeapi.co/api/v2/pokemon/1 (https://pokeapi.co/api/v2/pokemon) (http://api.tvmaze.com/shows/i) This endpoint returns an object that has all the details for a Pokémon with that ID:

- If the ID cannot be found in the Pokémon API (i.e. there is no Pokémon with that ID) you will return a 404 status code along with a "Pokémon Not Found!" error message.
- If the URL parameter is any other data type besides a positive whole number, you will respond with a 400 status code along with a "Invalid URL Parameter" error message.

NOTE: There are more Pokémon in the API than in the initial list that your http://localhost:3000/pokemon (http://localhost:3000/pokemon) route returns. This /:id route must work for EVERY valid ID in the Pokémon API not just the first 20 that are returned from the http://localhost:3000/pokemon (http://localhost:3000/pokemon) route.

NOTE: Remember, all URL parameters come through to the server as strings (even if they are numbers), you will need to do conversion to make sure they are positive whole numbers for the ID.

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.

Lab 5 Rubric

Criteria	Ratings		Pts
/pokemon Data Function and Route Endpoints will be tested with various edge cases	**25 to >0.0 pts** **Data Function and Route Endpoint** Data Functions and Route Endpoints are implemented correctly, and returns the data in correct type as per the lab requirements.	**0 pts** **All Test Cases Failed** Incorrect implementation, none of the test cases pass, or the function and route does not return anything.	25 pts
/pokemon/:id Data Function and Route Endpoints will be tested with various edge cases	**75 to >0.0 pts** **Data Function and Route Endpoint Implemented Correctly** Data Functions and Route Endpoints are implemented correctly, and returns the data in correct type as per the lab requirements.	**0 pts** **All Test Cases Failed** Incorrect implementation, none of the test cases pass, or the function and route does not return anything.	75 pts
			Total Points: 100