Tutorial 3

Express Server

To start off this tutorial, create two folders called frontend and backend. This will keep our frontend and backend logic separated, which is useful when projects get large.

In the backend folder, install the libraries express and cors by running npm install express cors. This will install the dependencies in a node_modules folder. It will also generate a package-lock.json that is used to keep track of your dependencies' versions.

Make a file called server.js. In this file we will setup an express server that handles getting, posting, and deleting tweets.

1. Create an array to store the tweets and create a currentId number variable and start it at 0.
2. Create a GET handler for the /tweets endpoint. This endpoint should return the array of tweets as JSON.
3. Create a POST handler for the /tweets endpoint. This endpoint should take in a tweet with the keys name, tweet and timestamp, add an id key to the object using the currentId variable, increment the currentId variable, then end the request.
4. Create a DELETE handler for the /tweets/:id endpoint. This endpoint takes in an id variable and deletes the tweet with the specified id from the array.

Note: if you try to hit these endpoints from a different origin (domain), the server won't let you as a pre-caution. In order to circumvent this, we installed the cors library. To use it, import the library by adding const cors = require('cors'); to the top of your file and add app.use(cors); somewhere underneath. This allows for Cross-Resource Origin Sharing. Read more about CORS here. (https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS)

Back to the frontend

Modify your frontend from the previous tutorial to use the new endpoints on the servers.

1. When the Tweet button is clicked, a POST request should be sent to the server using the fetch function with the new tweet. Upon success of the POST request, a subsequent GET request should be sent to the server to refresh the tweets. Note this will return an array of the tweets that includes the id in each object even though we did not send the id with our request.

2. Add a Delete button next to each tweet. Upon clicking this button, a DELETE request should be sent to the server to delete the tweet with the given id.