Putting it All Together: React Fetch CRUD Lab (CodeGrade)

- Due Mar 11 by 11:59pm
- Points 1
- Submitting an external tool
- Available until Mar 11 at 11:59pm

This assignment was locked Mar 11 at 11:59pm.

(https://github.com/learn-co-curriculum/react-hooks-fetch-crud-lab) (https://github.com/learn-co-curriculum/react-hooks-fetch-crud-lab/issues/new)

Learning Goals

- Use the `useEffect` hook to fetch data from an API
- Use event handlers with `fetch` to send data to an API
- Update state after receiving data from an API

Introduction

We're going to continue working on our quiz app, this time from the administrative side of things! We want our quizmasters to be able to view, add, edit, and delete the existing quizzes so they can change the content for the rest of our users.

There's some starter code set up, so you won't need to create any new components for this lab. Your goal will be to work with `fetch` and interact with our quiz API to perform the necessary CRUD operations to complete these deliverables.

Setup

As usual, make sure to run `npm install` to install the necessary dependencies.

For this lab, we'll be using `json-server` to create a RESTful API with our quiz data. You can run `json-server` by running `npm run server` (the command for this is in the `package.json` file). Once your server is running, go to `http://localhost:4000/questions` in your browser and have a look at the question data.

Then, in a new terminal tab, run `npm start` to run the React application.

Deliverables

GET /questions

When the application loads, get all the questions from `http://localhost:4000/questions` and display them using the `QuestionList` component.

You'll need to add `useState` and `useEffect` for this deliverable, but it's up to you to decide where it belongs! Think about which components will need access to the question data.

POST /questions

When the user clicks the 'New Question' button, a form will be displayed for creating a new question. This form is already set up as a controlled form, so your responsibility will be to send this form data to our API *when the form is submitted*.

For the API to work, you'll need to format your POST request like this:

```
POST /questions

Required Headers:
{ "Content-Type": "application/json" }

Body:
{
  "prompt": string,
  "answers": array of strings,
  "correctIndex": integer
}
```

In addition to updating the form, you should display the new question in the `QuestionList` component by updating state.

NOTE: because `json-server` doesn't have any validations, if you make any mistakes and send the body of your request in the wrong format, you'll need to manually delete the entry from the `db.json` file.

DELETE /questions/:id

When the user clicks the 'View Questions' button, a list of all the questions should show up (from deliverable 1). *When the delete button is clicked*, the question should be removed from the list by updating state. It should *also* be deleted on the server.

Make sure to include the **id** of the question you're trying to delete in your request's url!

PATCH /questions/:id

When the user clicks the 'View Questions' button, a list of all the questions should show up (from deliverable 1). *When the dropdown for the correct answer is changed*, the question should be updated on the server. It should also be updated in state.

For the API to work, you'll need to format your PATCH request like this:

```
PATCH /questions/:id

Required Headers:
{ "Content-Type": "application/json" }

Body:
{
  "correctIndex": integer
}
```

Make sure to include the **id** of the question you're trying to update in your request's url!

Resources

- [Using `fetch`](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch) (https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch)
- [React `fetch` with `useEffect` Example](https://reactjs.org/docs/faq-ajax.html#example-using-ajax-results-to-set-local-state) (https://reactjs.org/docs/faq-ajax.html#example-using-ajax-results-to-set-local-state)

- **React State and Arrays (https://github.com/learn-co-curriculum/react-hooks-state-arrays)**