React: CRUD

React – crud mini project

The following exercise contains the following subjects:

- HTTP
- CRUD

Submitting instructions:

- Push the full folder hierarchy of the project to your own repository on Github.

  Please add the following in hive:

  - A link to the repository
  - Free text – a description of the assignment. Stuff that you found hard to implement, known bugs and your review of this assignment

First things first

1. First get familiar with HTTP methods with axios in react
2. Get familiar with mock api

Our app

Lets create a react CRUD app with the mock api.

Get creative. We won't tell you what to create. You can create products like in amazon. Apartments like in airbnb etc.

I'll give an example of an avatar crud app below. This app will create, update, delete and read avatars that a user creates.

For simplicity sake the user can only create a name and an image for their avatar.

Do not mutate your state!!! Always create a copy of it. Here is how and why

Break it down

Lets start by thinking what we need to do. Lets start writing pseudo code and break down on how we will tackle this project step by step. Along the way I will discover bugs and think of ways to solve them.

Think like a React developer

- What kind of components should I have?
- Who should be the father and who should be the children?
- Which components I can reuse?
- What should I put in my state and what components should hold that state?

If there are components you find out that you can indeed reuse, don't sweat it. Just refactor that component so it can be reusable and implement it to your project.

Axios Setup

1) Install Axios as a dependency

2) Create an api.js file where we will create an axios baseURL to our mock api and export it. (So whenever we want to make a http request we can get reference to our baseUrl from the api.js file).

Random data

Lets play around with the mock api first and create some random data.

1) Lets create some random data in your mock api user interface.

2) Fetch the data when the component mounts to the screen.

3) Store the data to state

4) Display the data to the DOM

Create

Implement a create method that its job is to create avatars.

1) We will get the name and image url from the user.

2) Save the avatar to your state and also to your database.

While testing out your create method. You seem to find a bug. The user could submit their avatar without giving us a name or a valid image url. Lets correct that.

Lets create a few simple form validations. You check these validations before sending the ajax request.

3) User must give a name that is at least 5 characters long.

4) Image url must be a valid image url.

You discover another bug. You could create your avatar and click multiple times on the submit button which will produce multiple times the same avatar.

4) Disable the submit button when there is a request hanging (a request that is sent but no response received yet)

Update

Implement an Update method that will update a particular avatar in our state and in our database.

Delete

Implement a Delete method that will delete a particular avatar in our state and in our database

Spinner

Implement a loading spinner that will be displayed **whenever** you are making an ajax request.

React Router

Unfortunately we haven't learned yet about React router. So we can't navigate to different components. So we are not able to create links to individual items you created for example.

Submit the file to Hive.