

CSC 436: Web Applications

Lab 4

React Todo app

Goal: Manage global state using React Context and add data persistence to the React Todo App built in hw2 and hw3 by issuing REST calls (using Axios and the useResource hook) to a mock API (powered by json-server).

Requirements:

Update the your components to stop passing state and dispatch methods via props and instead create a StateContext, similar to the way demonstrated in class and then utilize then consume the StateContext in components which require access to state, dispatch, or both.'

Perform steps 2-7 to utilize a mock API to persist application data.

1. **Add StateContext**

- Create a StateContext in a contexts.js file.
- In App.js, wrap your components in a <StateContext.Provider> passing values for state and dispatch from your appReducer
- In App.js, remove state and dispatch props from your UserBar, CreateTodo, and TodoList components (your component names and structure may be different depending on how you implemented things thus far); instead, utilize the useContext hook within any component which may require access to state or dispatch.

2. **Create a Mock API using json-server**

Configure json-server and json-server-auth to run alongside your application using the db.json, setupProxy.js, and package.json startup commands. Reference Week 5's lecture for configuring these items.

Please reference the lecture for detailed instruction on how to utilize the files and commands to startup the mock API.

3. **Configure an Axios instance and RequestProvider in index.js** (See week 5 lecture)

4. **Use the useResource hook in order to retrieve Todos from the mock API and remove the hardcoded initialTodos list in app.js.** (See week 5/6 lecture)

5. **Use the useResource hook in order to issue a POST request to the mock API in order to persist your Todo to db.json.** (See week 6 lecture)

6. **Use the useResource hook in order to GET/POST requests to the mock in order to perform Login and Register.** (See week 6 lecture)

7. **Use the useResource hook in order to perform the appropriate HTTP request to update a Todo existing in db.json for TOGGLE_TODO and DELETE_TODO actions**

Remember, in order to accomplish all of the above you will need to install json-server and json-server-auth dependencies using NPM. It is strongly recommended that you create frontend/backend or client/server directories to organize and separate your frontend and



CSC 436: Web Applications

Lab 4

Submission

1. Add and push commits containing changes for lab 4 to the repo you use for lab 3. No need to worry if you have not yet received a grade for lab 3 prior to pushing, I will look at your commit history if need be.
2. Submit the link to your git repo to the lab 4 dropbox folder. This will indicate to me that your repo is ready for grading.

