Todo’s application

Description

Create a react application that uses api endpoints (provided) to create, delete, update and display Todos. Feel free to give the design of the UI based on your preference. The following items are required

1. The application should use **React Hooks, Redux with hooks** for state management.
2. Each Todo should display Title, Description
3. When Fetching todos, each todo should have a color. Display the actual color somewhere on the todo card/row, and not the text (the UI is up to you).
4. Expose buttons that will enable the user to:
    a. Create a todo - title, description and color of the todo should be required
    b. Delete a todo
    c. Update a todo - the user should be able to update all fields of a todo
5. The user should be able to clear the todo list (there is no delete all endpoint available, you should do this without “delete all” endpoint)
6. The todos should clear when the user closes the browser but should not clear on when the user reloads the window or closes the tab.

Endpoints

1. Get Todos
    a. **Method:** GET
    b. Url: https://todo.eachbase.com/api/[FirstnameLastname]/todos
    c. Request Body: Null
    d. Response Body:
        i. [{title: String, description: String, color: String}, ...]

2. Get Todo by Id
    a. **Method:** GET
    b. Url: https://todo.eachbase.com/api/[FirstnameLastname]/todos/:todoId
    c. Request Body: Null
    d. Response Body:
        i. {title: String, description: String, color: String}
3. Create
    a. **Method:** POST
    b. Url: https://todo.eachbase.com/api/[FirstnameLastname]/todos
    c. Request Body:
        i. { title: String, description: String, color: String }
    d. Response Body:
        i. {title: String, description: String, color: String}
4. Delete

    a. **Method:** DELETE
    b. **Url: https://todo.eachbase.com/api/[FirstnameLastname]/todos/:todoId**
    c. **Request Body: Null**
    d. **Response Body:**
        i. { id }

5. Update
    a. **Method:** PATCH
    b. **Url: https://todo.eachbase.com/api/[FirstnameLastname]/todos/:todoId**
    c. **Request Body:**
        i. { title: String, description: String, color: String }
    d. **Response Body**
        i. { title: String, description: String, color: String }

Example

The Urls will contain your first and last name as part of the path since more than one person may be accessing the API. Please only use the URL's we provide you. The following is an example url for a person with the name of John Smith for creating a new todo:

POST https://todo.eachbase.com/api/JohnSmith/todo/:todoId

Deployment

You can deploy your application anywhere you want (has to be live deployment with a url where we can go and test). However, you also have to submit your source code in a zipped folder via email.

If you have any questions feel free to reach us via email at admin@eachbase.com.

Thank you for taking the time to complete this challenge.Happy coding and don't forget to document your code!!!