

FED-Project 2: C(R)UD Operations for Books

Last commit into git: Monday, 24.06.2024 @ 9:45 a.m.

Presentation of UI and code: Monday, 24.06.2024 during your lab hour

1. **Create a custom hook:** In a new file, `src/domain/hooks.ts` create a custom hook function called `useBooks`. `useBooks` should automatically initiate fetching of the books **after the first render of your books' list** and return the currently available books, or an empty array until the server response was loaded. In addition to the array of your books (or an empty array on loading and on error), the function `useBooks` should also return the following:
 - `state` (of type `FetchState = "initial" | "loading" | "success" | "error"`)
 - `error` (of type `Error`)
 - `refresh` (a function)Display the state of fetching books in your App:
 - When `state === "loading"`, display "Loading books...."
 - When `state === "error"`, display the error message
 - When `state === "success"`, display the list of booksRefresh your list of books every 60 seconds (call the `refresh` function every 60 seconds). Make sure to stop fetching when the component is being destroyed. Replace the book fetching logic in your code by using your `useBooks` hook.
2. **Create Routes:** Add a navigation functionality to your app. Your user should be able to navigate between at least three screens, e.g.:
 - a. a screen displaying a list of books,
 - b. a "Add new book" screen, and
 - c. an "About" or "Imprint" screen.
 - d. Create an "Error" screen to handle errors
3. **Pagination:** Display only 10 (or any number up to 25) books per page, use the pagination endpoint of the Bookmonkey server: <http://localhost:4730/#operations>
4. **BookDetailScreen:** Add functionality to display all properties of a book on a separate screen. This screen should give the user the option to go back to "Books' list" screen.
5. **EditBookDetailScreen:** Add functionality to edit a book and save the changes in the backend. Editing a book must occur on a separate screen. After a successful "Update", display the `BookDetailScreen`. Add functionality to "Cancel and go back to `BookDetailScreen`" without changing the book.
6. **CreateBookScreen:** Add functionality to create a new book and save it to the backend. Creating a book must occur on a separate screen. After a successful "Save", display the `BookDetailScreen`. Add functionality to "Cancel and go back to "Books' list" screen".
7. **Delete Book functionality:** Add functionality to delete a book. After a successful delete, display the "books' list" screen
8. **For developers working in teams of two:**
 - In addition to 2), add routes for each book list category. Display only one list of books at a time. For users of your app, it must be apparent which list of books they are looking at and users must be able to easily switch to the other book list.
 - Both books list must use pagination, see 3).
 - Books in both lists must be viewable in detail, editable and deletable, see 4), 5) and 7)
 - Users must be able to create a new book, which, when created successfully, should appear in the correct, corresponding list of books.