‹ Back to React

MyReads: A Book Lending App

REVIEW | CODE REVIEW 8 | HISTORY

▾ src/App.js 3

```
import React from 'react'
import { Route } from 'react-router-dom'
import './App.css'
import SearchBooks from './SearchBooks'
import ListBooks from './ListBooks'
import * as BooksAPI from './BooksAPI'

class BooksApp extends React.Component {
    constructor(props) {
        super(props)
        this.updateBook = this.updateBook.bind(this)
    }

  state = {
```

SUGGESTION

If you take a look at React documentation it says that would be better to init the component state into the constructor.

```
    books: [],
    sbooks: [],
    query: ''
  }

  componentDidMount() {
```

AWESOME

It's very nice to know how to use `componentDidMount` idea, and you got it.

```
    this.refreshBooks()
    this.updateSearchBooks(this.state.books)
  }


  updateBook(book, shelf) {
    BooksAPI.get(book).then( book => {
      BooksAPI.update(book,shelf).then( () => {
        this.refreshBooks()
      })
    })
}

updateQuery = (query) => {
  this.setState({ query: query.trim() })
  this.updateSearchBooks(this.state.books)
}

updateSearchBooks(myBooks) {
  BooksAPI.search(this.state.query).then((sbooks) => {
    //first reset all search result books shelf to none
    sbooks.map( reset => reset.shelf = 'none' )
    sbooks.map( sb =>
      myBooks.map(mb =>
        ( sb.id === mb.id ? sb.shelf = mb.shelf : '' )
```

AWESOME

Nice job merging searched books with user's book

```
      )
    )
    this.setState({ sbooks })
```

```
  })
}

  refreshBooks() {
    BooksAPI.getAll().then((books) => {
      this.setState({ books })
      this.updateSearchBooks(books)
    })

  }

  render() {
    return (
      <div className="app">

        <Route exact path="/search" render={() => (
          <SearchBooks
            books={this.state.books}
            sbooks={this.state.sbooks}
            query={this.state.query}
            updateQuery={(query) => {this.updateQuery(query)}}
            onUpdateBook={(book, shelf) => {
              this.updateBook(book, shelf)

            }}
          />
        )}/>

        <Route exact path="/" render={() => (
          <ListBooks
            books={this.state.books}
            onUpdateBook={(book, shelf) => {
              this.updateBook(book, shelf)

          }}
          />
        )}/>

      </div>
    )
  }
}

export default BooksApp

```

▸ src/Book.js 2

▸ src/SearchBooks.js 1

▸ src/Bookshelf.js 1

▸ README.md 1

▸ src/index.js

▸ src/index.css

▸ src/ListBooks.js

▸ src/BookshelfControl.js

▸ src/BooksAPI.js

▸ src/App.test.js

▸ src/App.css

▸ public/index.html

▸ SEARCH_TERMS.md

▸ CONTRIBUTING.md

RETURN TO PATH