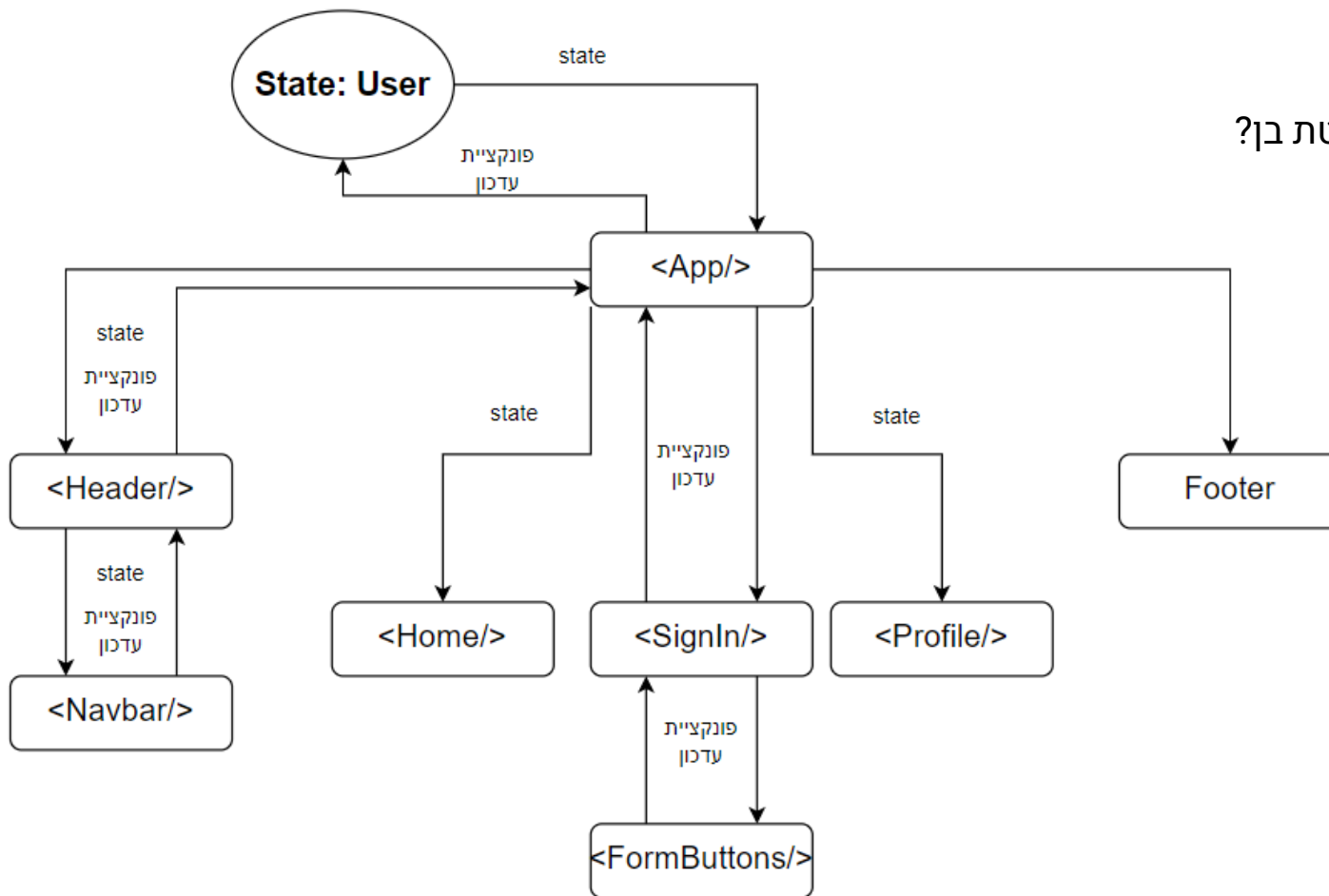


React + Ts

Redux & Global state management

01/09/2024

Props Drilling



- מה הוא עץ קומפוננטות?
- כיצד נוכל להעביר מידע מקומפוננטת אב לקומפוננטת בן?
- כיצד נוכל להכביר מידע בין קומפוננטות מרוחקות?
- מה משמעות המושג "props drilling"?
- מה הם החסרונות שבדבר?
- האם יש דרך להימנע מכך?

Redux וניהול state גלובלי – תרגול חלק א'

My App Home Sign In

Sign In

Email

"email" is not allowed to be empty

Password

...

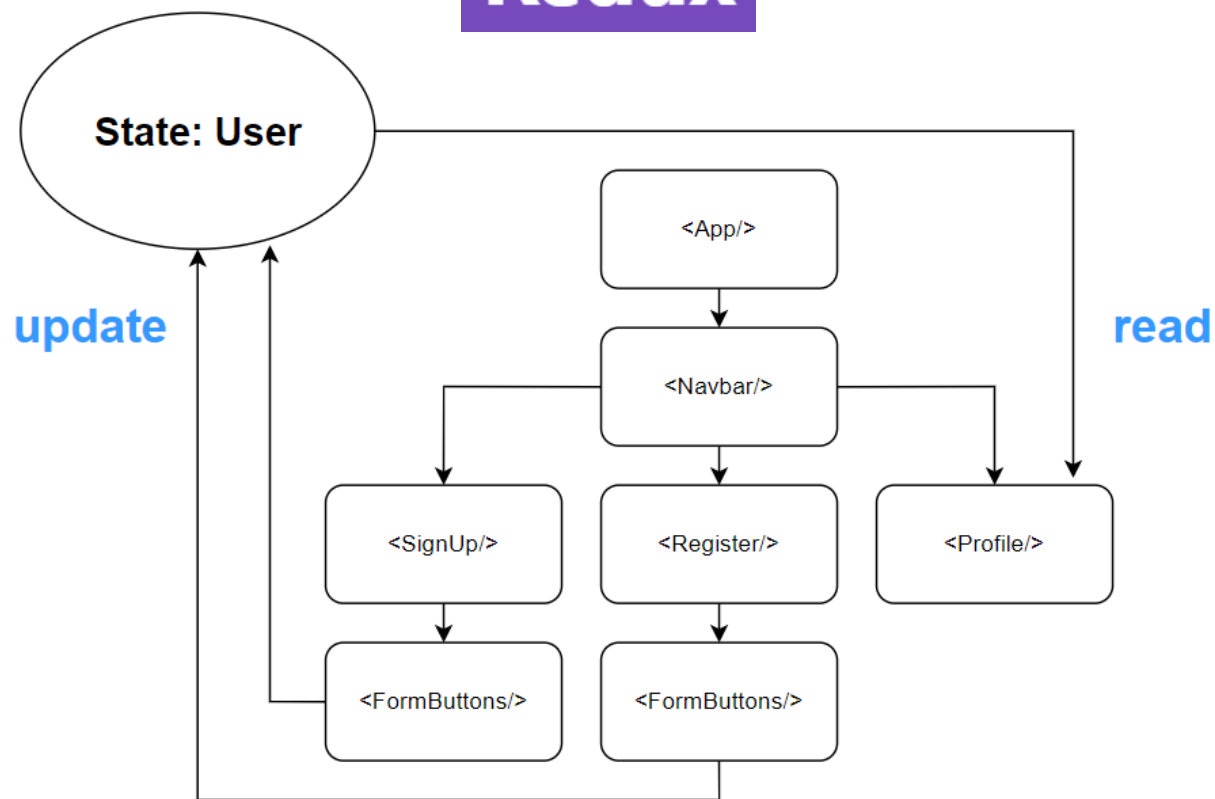
"password" must be at least nine characters long and contain an uppercase letter, a lowercase letter, a number and one of the following characters !@#\$%^&*-

Sign In

© 2024 Yehonatan Moravia

- בצעו fork לרפוזיטורי הבא:
- <https://github.com/yehonatan604/redux-lesson-starter>
- צרו state בוליאני בקומפוננטה App שיחזיק את המידע האם המשתמש מחובר או לא.
- בעת login מוצלח עדכנו state זה בהתאם.
- אל תציגו בבר הניווט את הקישור profile במידה ואין משתמש מחובר (בהתאם ל-state שייצרנו).
- במידה והמשתמש מחובר אל תציגו את הקישור לעמוד ההתחברות, הציגו כפתור התנתקות שבעת לחיצה עליו ה-state שייצרנו יעודכן בהתאם.

Redux



Redux וניהול state גלובלי

- מה זה ניהול state גלובלי?
- כיצד טכניקה זו תוכל לסייע לנו במניעת "props drilling"?
- מה היא Redux?
- מה היא @reduxjs/toolkit?
- מה נרצה לאחסן בתיקייה שניצור בשם Store?

Install

```
> npm i react-redux
```

Install

```
> npm i @reduxjs/toolkit
```

או

```
npm install @reduxjs/toolkit react-redux
```

Redux וניהול state גלובלי –

userSlice.ts

```
import { createSlice, PayloadAction } from "@reduxjs/toolkit";

const initialState = {
  isLoggedIn: false,
  userName: ""
}

const UserSlice = createSlice({
  name: "user",
  initialState,
  reducers: {
    login: (state: TUserState, data: PayloadAction<IUserLoginPayload>) => {
      state.isLoggedIn = true;
      state.userName = (data.payload).userName;
    },
    logout: (state: TUserState) => {
      state.isLoggedIn = false;
      state.userName = "";
    }
  }
});

export const userActions = UserSlice.actions;
export type TUserState = typeof initialState;
export type IUserLoginPayload = { userName: string; }
export default UserSlice.reducer;
```

- מה משמעות המושג ?slice
- איזה מידע חשוב לנו שיופיע ב-userSlice?
- לשם מה נשתמש במתודה createSlice?
- מה ניתן להעביר לאובייקט הקונפיגורציה של מתודה זו?
- מה נרצה להעביר כ-name?
- מה נרצה להעביר כ-initialState?
- מה זה reducer?
- איזה פונקציות נרצה להעביר כ-reducers?
- איזה פרמטרים נוכל להעביר לפונקציות אלו?
- מה זה payload?
- מדוע כדאי לנו ליצור type ל-initialState?
- כיצד נוכל לעשות זאת בצורה הכי קצרה ופשוטה שיש?
- למה נרצה לעשות export?

Redux וניהול state גלובלי – bigPie.ts

```
import { configureStore, combineReducers } from "@reduxjs/toolkit";
import userSlice from "../user.slice";

const store = configureStore({
  reducer: { userSlice },
  middleware: (getDefaultMiddleware) => {
    return getDefaultMiddleware({
      serializableCheck: false,
    });
  },
});

const rootReducer = combineReducers({ userSlice });
export type TRootState = ReturnType<typeof rootReducer>;
export default store;
```

- מה משמעות שם הקובץ bigPie.ts ?
- לשם מה נשתמש במתודה configureStore ?
- מה נוכל להעביר לה כאובייקט קונפיגורציה ?
- מדוע כדאי לנו להכין type ל-reducers ?
- כיצד נוכל לבצע זאת בקלות ?
- לשם מה נועדה התגית Provider ?
- איזה props יש לה שחשוב לנו להכיר ?
- מה נעביר ל-prop שנקרא "store" של תגית זו ?
- מה זה נותן לנו ?

```
ReactDOM.createRoot(document.getElementById("root")!).render(
  <React.StrictMode>
    <Provider store={bigPie}>
      <BrowserRouter>
        <App />
        <ToastContainer />
      </BrowserRouter>
    </Provider>
  </React.StrictMode>,
);
```

useSelector

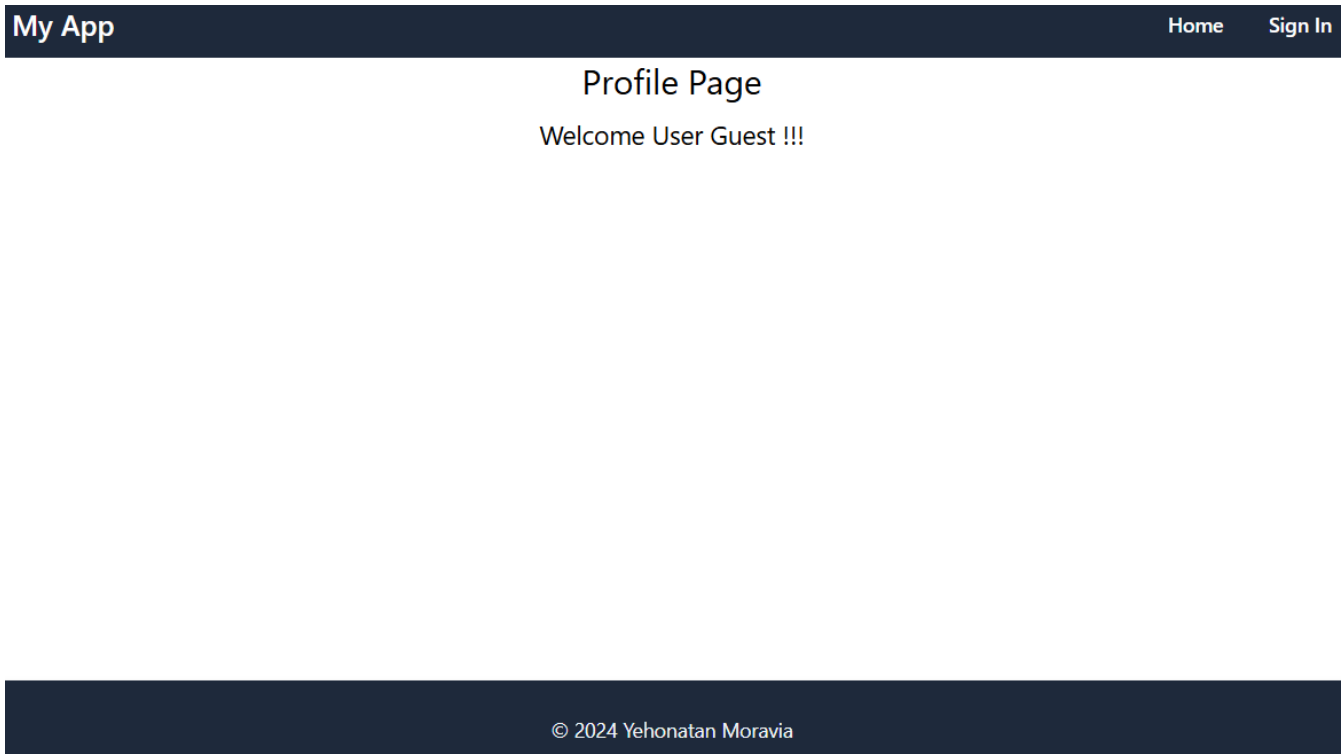
- לשם מה נשתמש ב-hook המיוחד useSelector?
- מה נעביר לו כפרמטר?
- מדוע אין לנו צורך לשמור אותו ב-state?

```
const Profile = () => {
  const user = useSelector(
    (state: TRootState) => state.userSlice,
  ) as TUserState;

  return (
    <div className="flex min-h-screen flex-col items-center justify-start gap-2">
      <h1 className="text-2xl">Profile Page</h1>
      <p className="text-lg dark:text-white">
        Welcome User {user.userName === "" ? "Guest" : user.userName} !!!{" "}
      </p>
    </div>
  );
};

export default Profile;
```

Redux וניהול state גלובלי – תרגול חלק ב'



- בצעו שוב fork לרפוזיטורי הבא:
<https://github.com/yehonatan604/redux-lesson-starter>
- צרו את התיקייה Store.
- צרו slice עבור המשתמש.
- צרו bigPie.
- הציגו את שם המשתמש בקומפוננטה profile, במידה ואין משתמש מחובר הציגו את השם Guest.
- אל תציגו בבר הניווט את הקישור profile במידה ואין משתמש מחובר.
- במידה והמשתמש מחובר אל תציגו את הקישור לעמוד ההתחברות, הציגו קישור התנתקות במקום.

useDispatch

```
const dispatch = useDispatch();
```

```
const submit = (form: { email: string; password: string }) => {
  console.log(form);
  dispatch(userActions.login({ userName: "John Doe" } as never));
};
```

- לשם מה נשתמש ב-hook המיוחד
?useDispatch
- מה נעביר כפרמטר לפונקציה
?dispatch

My App [Home](#) [Sign Out](#) [Profile](#)

Profile Page

Welcome User John Doe !!!



Profile Page

Welcome User Guest !!!

© 2024 Yehonatan Moravia

Profile Page

Welcome User John Doe !!!

© 2024 Yehonatan Moravia

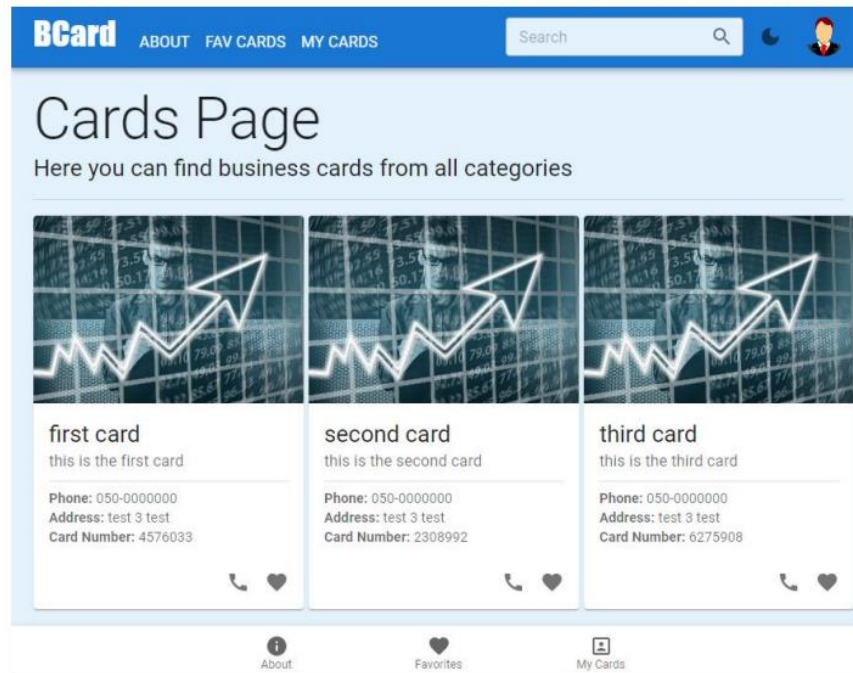
Redux וניהול state גלובלי – תרגול חלק ג'

- טפלו בכפתור של טופס ההתחברות כך שבעת שליחה יעודכנו פרטי המשתמש.
- טפלו בקישור ההתנתקות כך שיאפס את פרטי המשתמש.
- טפלו בתוכנית כך שהרשאת צפייה בעמוד הפרופיל תינתן למשתמש מחובר בלבד.
- טפלו בתוכנית כך שמשתמש מחובר לא יוכל לצפות בעמוד ה-login.

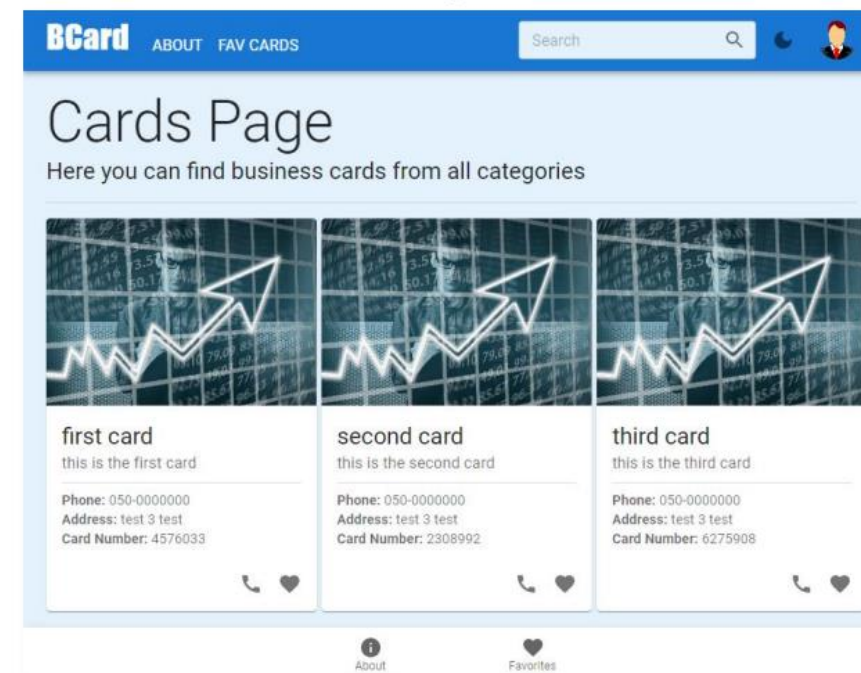
שיעורי בית

- יש לחזור ולסיים את כל התרגילים שבמצגת.
- פיתחו את פרויקט סוף המודול שלכם.
- התקינו את הספריות הדרושות לניהול state גלובלי באמצעות redux.
- טפלו בתוכנית ויישמו ניהול state גלובלי כפי שלמדנו בשיעור.
- במידה ויש משתמש מחובר הציגו לו בר ניווט בהתאם לנשפחים:

תפריט ניווט טעם משתמש מחובר מסוג עסקי



תפריט ניווט עם משתמש מחובר שאינו עסקי



תודה על ההקשבה

אני וצוות המכללה כאן עבורכם