import { createSlice } from "@reduxjs/toolkit";

const API = "https://65485be9dd8ebcd4ab22ca1b.mockapi.io/todos";

const TodoSlice = createSlice({

name: "todos",

initialState: {

todos: [],

name: "",

},

reducers: {

Set: (state, action) => {

return action.payload;

},

Add: (state, action) => {

const newTask = {

**id: Math.random(),**

name: action.payload,

complete: false,

};

state.todos.push(newTask);

fetch(API + "/" + state.name, {

method: "PUT",

headers: {

"Content-type": "Application/json",

},

body: JSON.stringify(state),

});

},

Delete: (state, action) => {

state.todos = state.todos.filter((item) => item.id !== action.payload);

fetch(API + "/" + state.name, {

method: "PUT",

headers: {

"Content-type": "Application/json",

},

body: JSON.stringify(state),

});

},

Update: (state, action) => {

state.todos = state.todos.map((item) => {

if (item.id === action.payload.id) {

return {

...item,

name: action.payload.name ? action.payload.name : item.name,

complete: action.payload.complete,

};

}

return item;

});

fetch(API + "/" + state.name, {

method: "PUT",

headers: {

"Content-type": "Application/json",

},

body: JSON.stringify(state),

});

},

},

});

export default TodoSlice.reducer;

export const { Add, Delete, Update, Set } = TodoSlice.actions;

import { configureStore } from "@reduxjs/toolkit";

import todosReducer from "./TodosSlice";

const store = configureStore({

reducer: todosReducer

});

export default store;  
import { useDispatch, useSelector } from "react-redux";

import { Add, Delete, Update, Set } from "../redux/TodosSlice";

const **Screen02** = ({ navigation, route }) => {

const API = "https://65485be9dd8ebcd4ab22ca1b.mockapi.io/todos";

useEffect(() => {

fetch(API + "?name=" + route.params.name)

.then((response) => response.json())

.then((data) => {

if (data[0]) {

console.log(data[0]);

dispatch(Set(data[0]));

} else {

fetch(API, { method: "POST", headers: {

"Content-type": "Application/json",

},

body: JSON.stringify({

userName: route.params.name,

todos: [],

}),

})

.then((res) => res.json())

.then((data) => dispatch(Set(data)));

}

});

}, []);

const [newTask, setNewTask] = useState("");

const todos = useSelector((state) => state.todos);

const **dispatch** = useDispatch();

**const Item = ({ item }) => {**

const [taskUpdate, setTaskUpdate] = useState("");

let isComplete = item.complete;

**const handleUpdate** = () => {   
 dispatch( Update({ id: item.id,name: taskUpdate,

complete: isComplete, }) ); };

**return** ( <View style={{ }}>

<Text style={{ width: "100%" }}>{**item.name**}</Text>

<**CheckBox** value={isComplete} onValueChange={() => {

isComplete = !isComplete; handleUpdate(); }} />

<Button title="**Delete**"onPress={() => {dispatch(Delete(item.id)); }} />

<Button title="**Udpate**"

onPress={() => { **handleUpdate**(); setTaskUpdate(""); }} />

</View>

<TextInput value={**taskUpdate**} onChangeText={(text) => setTaskUpdate(text)} placeholder="update here" />

</View>

);

};

**return** **(**

<Text>User: {route.params.name}</Text>

<ScrollView>

{todos.map((item) => {

return <Item key={item.id} item={item} />; })}

</ScrollView>

<Button title="Add" onPress={() => { dispatch(Add(newTask)); setNewTask(""); }} />

<TextInput value={**newTask**} onChangeText={(text) => setNewTask(text)} placeholder="Add here" />

**)**

-------------------------------------------------------------------------

import { NavigationContainer } from '@react-navigation/native';

import { createNativeStackNavigator } from '@react-navigation/native-stack';

import { Provider } from 'react-redux';

import store from './redux/store';

const stack = createNativeStackNavigator();

<Provider store={store}> <NavigationContainer> <stack.Navigator>

<stack.Screen name="Screen01" component={Screen01} />

</stack.Navigator> </NavigationContainer> </Provider>

git config --global user.name "John Doe"

git config --global user.email johndoe@example.com

git config --list

git init

git add .

git commit -m "ghi gi thi gi"

"@expo/webpack-config": "^19.0.0",

"@react-navigation/native": "^6.1.9",

"@react-navigation/native-stack": "^6.9.16",

"expo": "~49.0.15",

"expo-status-bar": "~1.6.0",

"react": "18.2.0",

"react-dom": "18.2.0",

"react-native": "0.72.6",

"react-native-safe-area-context": "^4.6.3",

"react-native-screens": "~3.22.0",

"react-native-web": "~0.19.6",

"@reduxjs/toolkit": "^1.9.7",

"react-redux": "^8.1.3",

"redux": "^4.2.1"