apsara

Date:

ep-12 Let's Build Our App

- Managing state of application using redux. (Data managment)
- Redux works on the data layer.
- Redux is state managment library

Note: Redux is not mandatory.

- redux & react are different libraries
- Zustand is another library like redux to manage state
- Redux offers easy debugging

→ Redux

- Redux also works with other libraries or framework
- But it is most popularly used with react
- offers 2 libraries

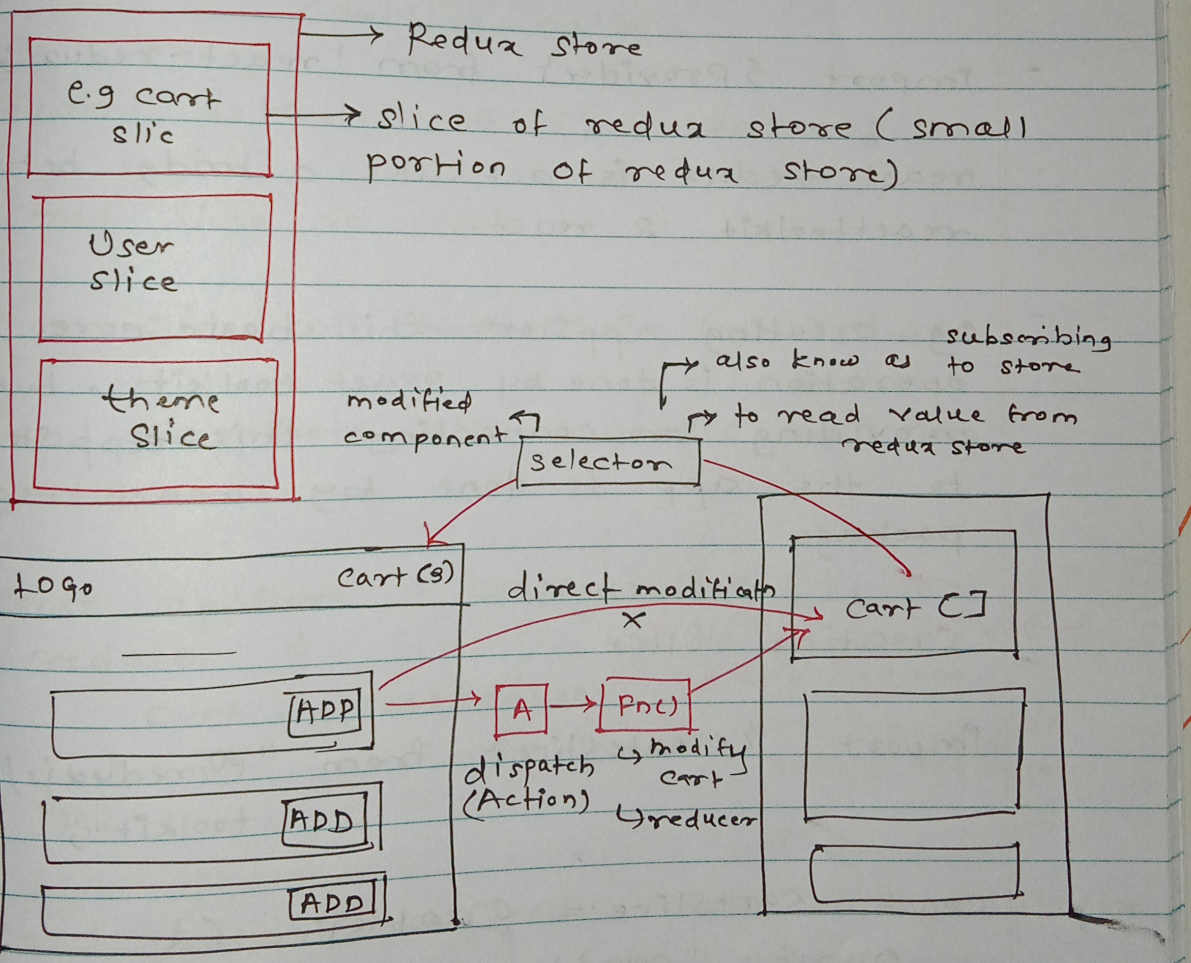
1) React - Redux
2) Redux Toolkit → Newer way of writing redux

↳ Redux Toolkit

- Feature - Cart Flow (Adding items to cart) To store all cart information we will be using redux

↳ Architecture of Redux Toolkit (RTK)

Redux store - It is a big object and kept in central global place. It have multiple data in it



When we press the add button, it dispatches an action, which calls the reducer function that updates the slice of redux store

Installation

1] npm install @reduxjs/ toolkit
2] npm install react-redux.

creating appStore

e.g const appStore = configureStore ({});

export default appStore;

Providing appStore to our application

- import {Provider} from 'react-redux';
- react-redux is act like a bridge between reactToolkit & react.
- e.g creating appStore this basic core operation is done by react toolkit, but providing or connecting this appStore to the app is done by react-redux package

creating Slice

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

```
const cartSlice = createSlice ({
    name: 'cart',
    initialState : {
        items : []
    },
```

```
reducers : {
        addItem : (state, action) => {
            state.items.push(action.payload);
        },
        removeItem : (state) => {
            state.items.pop();
        }
    }
})

export const { addItem, removeItem } =
                cartSlice.actions

export default cartSlice.reducer;
```

↳ Add Slice to store

- Each slice will have their own reducer.
- and appStore also have its own big reducer

eg
```
const appStore = configureStore({
    reducer: {
        cart : cartReducer,
    },
});
```

app Reducer — combination of slice's reducer

↳ Read value from redux store using selector

Selector is a hook inside react

e.g const cartItems = useSelector ((store)) =>
store.cart.items);

useSelector() hook is given by react-redux

↳ dispatch an action

eg const dispatch = useDispatch();

const handleClearCart = () => {
dispatch (clearCart());
}

<button onClick={handleCart}>Clear
cart</button>

↳ Careful while using selectors

eg 1 const store = useSelector((store) => store);

const cartItems = store.cart.items;

eg 2 const cartItems = useSelector((store) =>
store.cart.items);

e.g 1 is less efficient than e.g 2.

because in eg 2 we are subscribing specific portion of the store not whole store

apsara

Date: __.__.____

↳ reducer vs reducers

- reducer in appStore which have multiple combination of slice reducers
- reducers are combination of multiple dispatch functions in every slice we created.