**Day 3 :21 Jul. 24**

**useState()** : This hook is use to create state variable in functional components.

**useReducer()** : if we want to write any complex logic for particular component then userReducer() hook responsible to provide external functionality using reducer.

Context API : Context API is a part of react js which provided two hook **createContext()** and **useContext()** which help to share the data between one component to another component doesn’t matter their relationship on n level.

This concept is use full when we are developing small level application.

Redux is a third party library which help to make local state variable as global variable. Redux also known as predictable state container for JavaScript.

JavaScript with Redux

React with Redux

Angular with Redux NgRx

Redux Design pattern

3 core principles

1. Single source of truth ie store
2. State in ready only
3. Change for state variable part of store using Pure JavaScript function ie reducer.

Key concept

1. store : it is use to store the variable of any types.
2. reducer : Pure JavaScript function takes 2 parameter initial state and action
3. dispatch : with help of dispatch we can pass action as well as payload. Using action we do the changes on state variable part of redux store.
4. action : action is uses to do the change on state variable.
5. state : consider as global variable with help to store

react with redux with class object we are/were using connect HOC concept to connect reducer.

From react with hook they provided two hook ie

useSelector() : it is use to access global state variable in each components. Doesn’t matter their relationship.

useDispatch() : it use to pass the action and payload to reducer function. Reducer function do the state on state variable part of store base upon type of actions.

create-react-app react-with-redux-employee-app

cd react-with-redux-employee-app

npm install redux react-redux

RTK (Redux Tool kit) : RTK is the official recommended way to write the redux logic in simple way. RTK provided easier tool and best practise to achieve redux logic.

Limitation of core Redux

1. Set up : reducer function, actions and store configuration.
2. Reducer function : we need to check the action using if state or switch statement and base upon action we do changes on state.
3. Immutability : all variable part of store are immutable variable while doing some changes we can get some error.
4. If we need any middle ware is logger, browser plugin , redux with thunk( help to achieve asyn call) etc.
5. Best practises

RTK

1. Setup : createStore is replace by configureStore function. createSlice function which contains initial state, reducer functionality etc.
2. Reducer : use createSlice which help to generate action creation and action types.
3. It provide in build redux with thunk middle ware plugin to handle asyn call.

create-react-app react-with-redux-toolkit-counter-app

cd react-with-redux-toolkit-counter-app

npm install @reduxjs/toolkit react-redux

React redux thunk and react redux saga

Both are third party library which also known as middleware for the redux application. Which help to interact with Rest API asynchronous communication.

If we are doing simple asynchronous communication with backend technologies then we can use redux with thunk

If we want to use complex asynchronous call. Then we can use redux saga middleware which is base upon ES6 generator functions.

Core redux with react then we need to install redux-thunk middleware.

npm install redux react-redux redux-thunk axios

React toolkit internally use redux-thunk.

npm install @reduxjs/toolkit react-redux axios

create-react-app react-redux-toolkit-product-app

cd create-react-app react-redux-toolkit-product-app

npm install @reduxjs/toolkit react-redux axios