[08.12] 리액트 리덕스

React-Redux

Redux 상태 관리 라이브러리

-전역 상태(state)를 중앙 집중식으로 관리하기 위한 패턴과 라이브러리.

-전역 상태 관리

-예측 가능

-디버깅 용이

React-Redux는 Redux를 React 컴포넌트와 연결

[설치]

npm install redux react-redux

npm install @reduxjs/toolkit

기본 구성요소

1. store: 전체 앱의 상태 트리 보관소 (Redux의 중심)

2. reducer: 상태(state)를 변경하는 함수

3. action: 상태를 변경하기 위한 명령 (객체)

4. Provider: store를 React 컴포넌트에 주입

5. useSelector: store에서 값을 읽어오는 훅

6. useDispatch: action을 dispatch하는 훅

[counterSlice.jsx] (reducer + actions)

// Redux Toolkit을 쓰면 reducer + action 생성이 간편해짐

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

const counterSlice = createSlice({

name: 'counter',

initialState: { value: 0 },

reducers: {

increment: (state) => { state.value += 1; },

decrement: (state) => { state.value -= 1; },

},

});

export const { increment, decrement } = counterSlice.actions;

export default counterSlice.reducer;

[store.jsx]

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

import counterReducer from './counterSlice';

const store = configureStore({

reducer: {

counter: counterReducer,

},

});

export default store;

[main.jsx] (Provider 연결)

import { StrictMode } from 'react'

import { createRoot } from 'react-dom/client'

import './index.css'

import App from './App.jsx'

import { Provider } from 'react-redux';

import store from './store';

//<Provider store={store}>: store를 전역에 공급

createRoot(document.getElementById('root')).render(

<StrictMode>

<Provider store={store}>

<App />

</Provider>

</StrictMode>,

)

[App.jsx]

import React from 'react';

import { useSelector, useDispatch } from 'react-redux';

import { increment, decrement } from './counterSlice';

function App() {

//useSelector(): store에서 state 읽기

const count = useSelector((state) => state.counter.value);

const dispatch = useDispatch();

return (

<div>

<h1>Count: {count}</h1>

<button onClick={() => dispatch(increment())}>증가</button>

<button onClick={() => dispatch(decrement())}>감소</button>

</div>

);

}

export default App;

Redux Toolkit 소개

-Redux를 더 쉽게 쓰도록 돕는 공식 도구

-createSlice()로 reducer + action 생성

-configureStore()로 store 설정 간단화

-Immer 내장 → 불변성 처리 자동

-thunk 내장 → 비동기 처리 쉬움

핵심 hook

useSelector: Redux store에서 state를 읽음

useDispatch: action을 dispatch함

useStore: store 전체 객체 접근 (거의 안 씀)

