



## • zustand 요약



# 1. zustand 소개

## ❖ zustand 란?

- zustand : 독일어로 상태라는 뜻
- 작고 가볍고 확장성이 뛰어난 상태 관리 라이브러리
- Hook 기반의 API 제공
- flux 아키텍처를 단순화시켰음

## ❖ zustand의 가장 큰 특징

- 다중 스토어
  - 다중 스토어 : 스토어를 여러개를 만들어서 필요한 곳에 Injection 하여 사용
  - Redux는 단일 스토어 --> 상태 변경 기능을 다중 리듀서로 분리했었음
- 상태와 액션의 구분이 없음
  - 상태를 변경하는 액션도 넓은 의미에서 상태로 간주함
- 간편한 비동기 처리
  - 액션 메서드에서 직접 비동기 처리함
  - Redux 처럼 복잡하지 않음



## 2. zustand 설치 및 사용

### ❖zustand 설치

- npm install zustand
- npm instal immer : 어차피 쓸거니까 미리 추가

### ❖store 형태 : 복잡한 객체형태일 때는 불변성을 가지도록 변경

```
import { create } from "zustand";

const stateCreator = (set) => ({
  count1: 0,
  nested: { count2: 0 },
  inc1: () =>
    set((state) => ({
      count1: state.count1 + 1,
    })),
  inc2nested: () =>
    set((state) => ({
      nested: { ...state.nested, count2: state.nested.count2 + 1 },
    })),
});

const useCountStore = create(stateCreator);

export default useCountStore;
```

쉽지 않다..

## 2. zustand 설치 및 사용

### ❖ 컴포넌트에 주입시킬 때

```
import useCountStore from "../store/useCountStore";

export default const App = () => {
  const count1 = useCountStore((state) => state.count1);
  const nested = useCountStore((state) => state.nested);
  const inc1 = useCountStore((state) => state.inc1);
  const inc2nested = useCountStore((state) => state.inc2nested);

  return (
    <div className="App">
      count1 : {count1}
      <br />
      count2 : {nested.count2}
      <br />
      <button onClick={() => inc1()}>count1 증가</button>
      <button onClick={() => inc2nested()}>count2 증가</button>
    </div>
  );
};
```

### 3. immer 사용

#### ❖zustand immer 미들웨어

- npm install immer
- immer의 draft 를 이용하듯이 변경함. 새로운 상태를 리턴할 필요 없음

```
import { create } from "zustand";
import { immer } from "zustand/middleware/immer";

const stateCreator = (set) => ({
  count1: 0,
  nested: { count2: 0 },
  inc1: () =>
    set((state) => {
      state.count1 = state.count1 + 1;
    }),
  inc2nested: () =>
    set((state) => {
      state.nested.count2 = state.nested.count2 + 1;
    }),
});

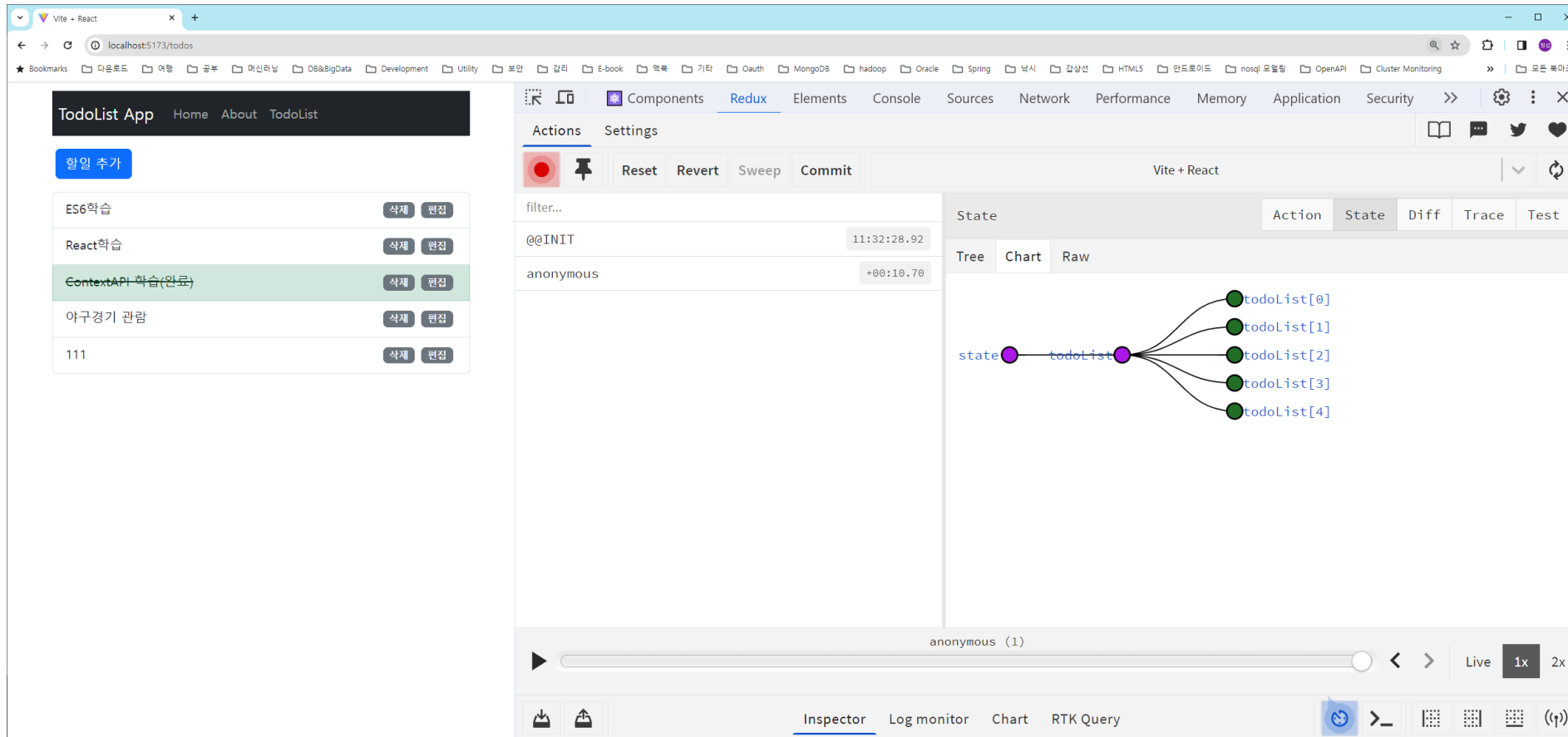
const useCountStore2 = create()(immer(stateCreator));

export default useCountStore2;
```

## 4. devtools

### ❖ redux devtools

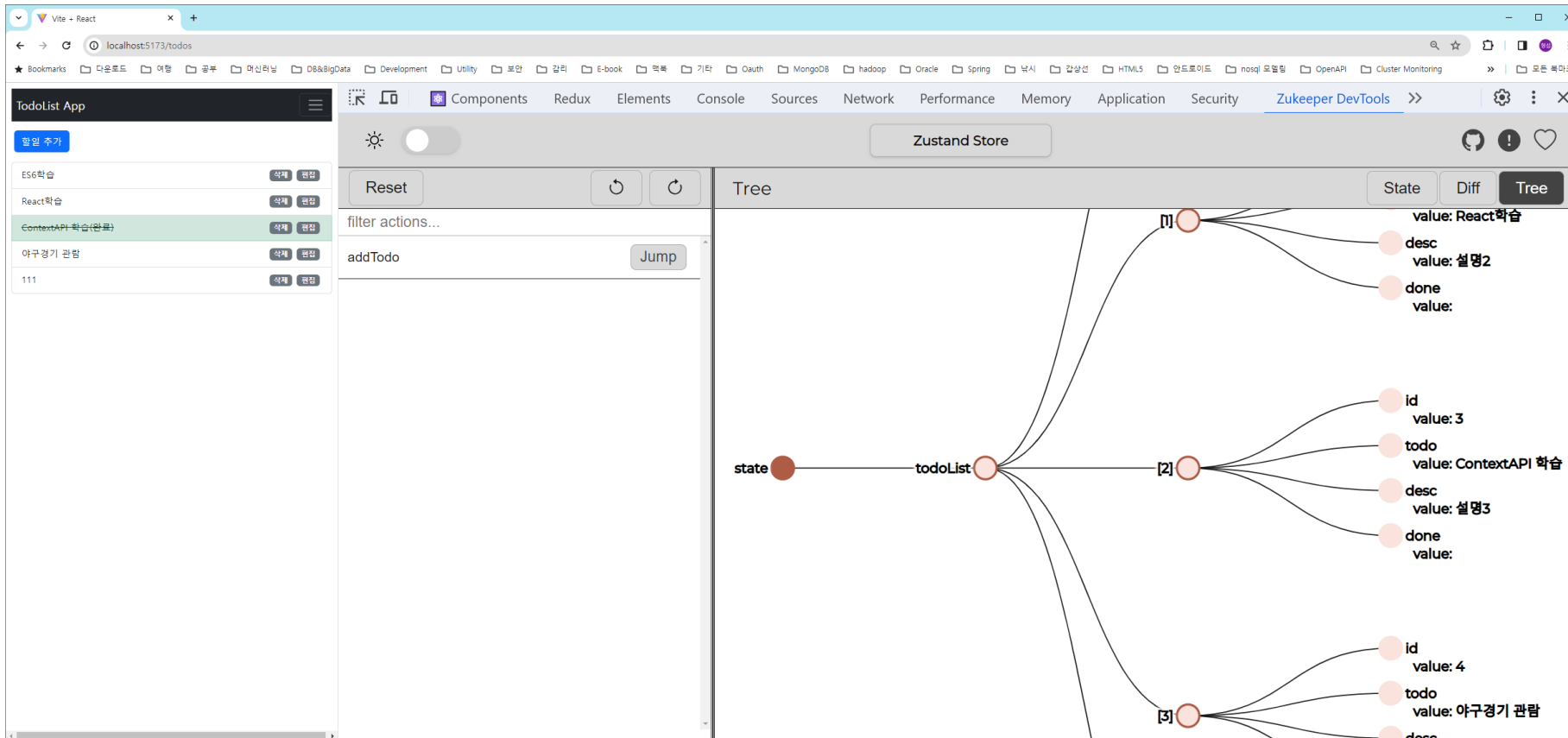
- zustand 스토어의 상태를 redux devtools로 관찰할 수 있도록 지원



## 4. devtools

### ❖zookeeper

- <https://www.zookeeper-tools.com/>
- zustand 전용의 devtools. 출시된지 얼마되지 않아 약간의 버그가 눈에 띈다





## 5. 비동기 처리

❖ 액션 메서드에서 바로 비동기 처리 : contacts-client-zustant-2 예제 참조

```
import axios from "axios"
import { create } from "zustand";

const BASEURL = "https://contactsvc.bmaster.kro.kr/contacts_long/search/";

const stateCreator = (set)=>({
  contacts : [],
  isLoading : false,
  asyncSearchContacts : async (name, errorCallback)=> {
    set((state)=>({ contacts: [], isLoading: true}));
    try {
      const response = await axios.get(BASEURL + name);
      set((state)=>({ contacts: response.data }));
    } catch (e) {
      errorCallback("에러발생 : " + e.message);
    }
    set((state)=>({ isLoading: false }));
  }
})

const useContactStore = create(stateCreator);

export default useContactStore;
```





Q&A