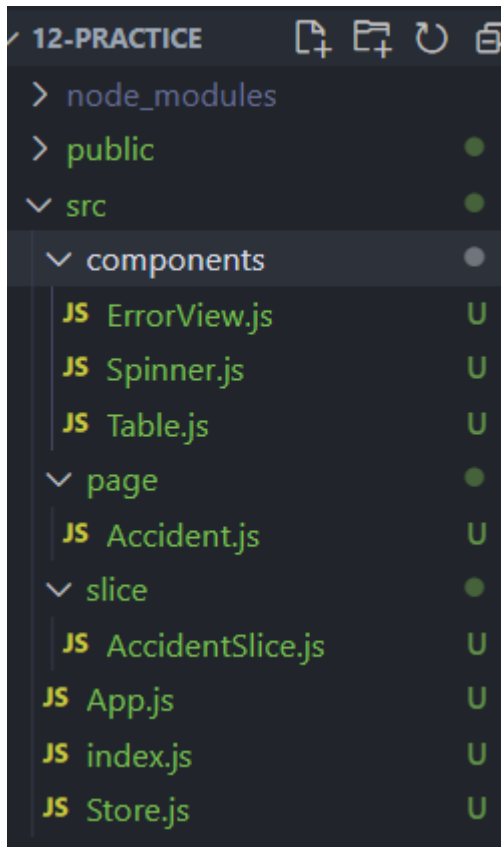


# Redux- 교통사고 현황 연습 문제 - 박세영

---

## DOM구조



---

## App.js

```
import React from 'react';
import Accident from './page/Accident';

function App() {
  return (
    <div>
      <h1> Redux 연습문제 </h1>
      <hr />
      <br/>
      <Accident/>
    </div>
  );
}

export default React.memo(App);
```

## AccidentSlice.js

```
import {createSlice, createAsyncThunk} from '@reduxjs/toolkit';
import axios from 'axios';

//비동기 처리 함수 구현- ajax요청을 위한 함수
export const accidentList= createAsyncThunk('accident/accidentList',
async(payload, {rejectWithValue})=>{
  let result= null;
  try{
    result = await axios.get('http://localhost:3001/traffic_acc',{
      params:{
        //payload로 params전달..
        year: payload?.year
      }
    });
  }catch(e){
    result= rejectWithValue(e.response)
  }
  return result; //ajax 연동 결과를 리턴
});

//slice 정의 (action함수+ reducer의 개념)
//ajax의 처리결과를 관리할 상태값 정의
const accidentSlice= createSlice({
  name: 'accident',
  initialState:{
    data: null, //ajax처리를 통해 수신된 데이터
    loading: false, //로딩 여부
    error: null //에러 정보
  },
  extraReducers:{ // ajax연동기능을 호출할때 사용
    [accidentList.pending]: (state, {payload})=>{ //로딩중-기존의 상태값
      복사후 로딩만 true로 바꿈
      return {...state, loading: true}
    },
    [accidentList.fulfilled]: (state, {payload})=>{ //ajax완료-성공
      return{
        data: payload?.data, //payload가(axaj결과가) 존재할때만 data에 접
        loading: false,
        error: null
      }
    },
    [accidentList.rejected]: (state, {payload})=>{ //에러
      return{
        data: payload?.data,
        loading: false,
        error:{
          code: payload?.status ? payload.status : 500,
          message: payload?.statusText ? payload.statusText : 'Server
Error'
```

```

    }
  }
}
});

//리듀서 객체 내보내기
export default accidentSlice.reducer;

```

## Store.js

```

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

import accidentSlice from './slice/AccidentSlice';

const store= configureStore({
  reducer: {
    accident: accidentSlice
  },
  middleware: [...getDefaultMiddleware({serializableCheck: false})],
  devtools: true
});

export default store;

```

## Accident.js

```

import React, { memo } from "react";
import styled from "styled-components";

import Spinner from "../components/Spinner";
import Table from "../components/Table";
import ErrorView from "../components/ErrorView";

//상태값을 로드하기위한 hook과 action함수를 dispatch할 hook 참조
import { useSelector, useDispatch } from "react-redux";
//slice에 정의된 액션 함수들 참조
import { accidentList } from "../slice/AccidentSlice";

//드롭다운을 배치하기 위한 박스
const SelectContainer = styled.div`
  position: sticky;
  top: 0;
  background-color: #fff;
  border-top: 1 solid #eee;
  border-bottom: 1px solid #eee;
  padding: 10px 0;

```

```

margin: 0;
select {
  margin-right: 15px;
  font-size: 16px;
  padding: 5px 10px;
}
`;

const Accident = memo(() => {
  React.useEffect(() => console.clear(), []);
  const { data, loading, error } = useSelector((state) => state.accident);
  const [year, setYear] = React.useState();
  const dispatch = useDispatch();

  let sumAccident = 0;
  let sumDeath = 0;
  let sumInjury = 0;

  React.useEffect(() => {
    dispatch(accidentList({year: year}));
  }, [dispatch, year]);
  const onSelectChange = React.useCallback((e) => {
    e.preventDefault();
    //드롭다운 입력값 취득
    const current = e.target;
    const value = current[current.selectedIndex].value;
    setYear(value);
  }, []);

  return (
    <div>
      <Spinner visible={loading} />
      <SelectContainer>
        <select name="year" onChange={onSelectChange}>
          <option value=""> -- 년도 선택 -- </option>
          {[...new Array(2018 - 2005 + 1)].map((v, i) => {
            return (
              <option key={i} value={2005 + i}>
                {2005 + i}년도
              </option>
            );
          })}
        </select>
      </SelectContainer>
      {error ? (
        <ErrorView error={error} />
      ) : (
        <Table>
          <thead>
            <th>번호</th>
            <th>년도</th>
            <th>월</th>
            <th>교통 사고 건수</th>
            <th>사망자 수</th>
          </thead>
        </Table>
      )}
    </div>
  );
});

```

```

    <th>부상자 수</th>
  </thead>
  <tbody>
    {data &&
      data.map(({ id, year, month, accident, death, injury }, i) => {
        sumAccident += accident;
        sumDeath += death;
        sumInjury += injury;
        return (
          <tr key={id}>
            <td>{id}</td>
            <td>{year}년</td>
            <td>{month}월</td>
            <td>{accident}건</td>
            <td>{death}명</td>
            <td>{injury}명</td>
          </tr>
        );
      })}
    <tr key="sum" style={{ fontWeight: "bolder" }}>
      <td
        colspan="3"
        style={{
          color: "white",
          border: "3px solid #168",
          backgroundColor: "#168",
        }}
      >
        합계
      </td>
      <td style={{ border: "3px solid #168" }}>{sumAccident} 건</td>
      <td style={{ border: "3px solid #168" }}>{sumDeath} 명</td>
      <td style={{ border: "3px solid #168" }}>{sumInjury} 명</td>
    </tr>
  </tbody>
</Table>
  )}
</div>
);
});

export default Accident;

```

---

components 폴더안의 파일들과 index.js는 이전 제출물과 같습니다.

---

## 출력 결과

### 초기 화면

# Redux 연습문제

-- 년도 선택 -- ▾

번호	년도	월	교통 사고 건수	사망자 수	부상자 수
1	2005년	1월	15494건	504명	25413명
2	2005년	2월	13244건	431명	21635명
3	2005년	3월	16580건	477명	25550명
4	2005년	4월	17817건	507명	28131명
5	2005년	5월	19085건	571명	29808명
6	2005년	6월	18092건	476명	28594명
7	2005년	7월	18675건	528명	29984명
8	2005년	8월	19035건	562명	31603명
9	2005년	9월	18759건	577명	29831명
10	2005년	10월	19757건	639명	31597명
11	2005년	11월	19129건	574명	30337명
12	2005년	12월	18504건	530명	29750명
13	2006년	1월	14971건	420명	24533명
14	2006년	2월	14270건	373명	22903명
15	2006년	3월	16767건	465명	26013명
16	2006년	4월	17948건	469명	28725명
17	2006년	5월	19140건	531명	30279명
18	2006년	6월	17435건	455명	27032명
19	2006년	7월	18634건	516명	29978명
20	2006년	8월	18794건	585명	30882명
21	2006년	9월	19293건	580명	30186명
22	2006년	10월	19100건	651명	30715명
23	2006년	11월	19877건	701명	31270명
24	2006년	12월	17516건	581명	27713명
25	2007년	1월	14014건	400명	22075명

6 / 12

25	2007년	1월	14914건	468명	23915명
26	2007년	2월	14696건	446명	23717명
27	2007년	3월	18166건	476명	28811명
28	2007년	4월	18055건	460명	28555명
29	2007년	5월	19264건	516명	30532명
30	2007년	6월	18310건	538명	28662명
31	2007년	7월	18037건	503명	28676명
32	2007년	8월	17982건	512명	29711명
33	2007년	9월	18506건	532명	29583명
34	2007년	10월	19026건	582명	29802명
35	2007년	11월	17815건	565명	27674명
36	2007년	12월	16891건	568명	26208명
37	2008년	1월	14741건	444명	23522명
38	2008년	2월	14176건	428명	23282명
39	2008년	3월	16752건	502명	26284명
40	2008년	4월	17498건	476명	27364명
41	2008년	5월	19515건	493명	30935명
42	2008년	6월	18824건	423명	28872명
43	2008년	7월	19562건	452명	30553명
44	2008년	8월	18545건	529명	29705명
45	2008년	9월	18023건	502명	27894명
46	2008년	10월	19926건	574명	30882명
47	2008년	11월	19597건	528명	30585명
48	2008년	12월	18663건	519명	29084명
49	2009년	1월	16532건	475명	27373명
50	2009년	2월	15502건	405명	24429명
51	2009년	3월	18224건	468명	27967명
52	2009년	4월	19153건	439명	29521명
53	2009년	5월	20237건	519명	31743명
54	2009년	6월	19249건	433명	29084명

55	2009년	7월	20281건	454명	31545명
56	2009년	8월	19993건	514명	32200명
57	2009년	9월	21004건	500명	31700명
58	2009년	10월	21440건	592명	33255명
59	2009년	11월	20156건	522명	31062명
60	2009년	12월	20219건	517명	31996명
61	2010년	1월	17124건	438명	27655명
62	2010년	2월	15803건	406명	24968명
63	2010년	3월	17801건	396명	27350명
64	2010년	4월	18386건	395명	28191명
65	2010년	5월	20372건	447명	31521명
66	2010년	6월	18910건	474명	28311명
67	2010년	7월	18965건	453명	29577명
68	2010년	8월	19383건	410명	31032명
69	2010년	9월	19439건	436명	30470명
70	2010년	10월	21575건	619명	33282명
71	2010년	11월	20271건	541명	30907명
72	2010년	12월	18849건	490명	29194명
73	2011년	1월	16454건	395명	25611명
74	2011년	2월	14208건	339명	22493명
75	2011년	3월	16832건	338명	25371명
76	2011년	4월	18570건	429명	28534명
77	2011년	5월	19414건	393명	29906명
78	2011년	6월	18505건	415명	28469명
79	2011년	7월	19253건	462명	29576명
80	2011년	8월	18896건	472명	30034명
81	2011년	9월	19065건	474명	29055명
82	2011년	10월	20952건	520명	32133명
83	2011년	11월	20283건	514명	30917명
84	2011년	12월	19279건	478명	29292명



85	2012년	1월	16818건	418명	26281명
86	2012년	2월	16656건	393명	25998명
87	2012년	3월	18255건	403명	27899명
88	2012년	4월	19372건	483명	29628명
89	2012년	5월	19672건	444명	30163명
90	2012년	6월	18854건	476명	28314명
91	2012년	7월	19333건	416명	29642명
92	2012년	8월	18407건	409명	29051명
93	2012년	9월	19234건	486명	29462명
94	2012년	10월	19557건	533명	29759명
95	2012년	11월	19750건	508명	30054명
96	2012년	12월	17748건	423명	28314명
97	2013년	1월	16016건	389명	25094명
98	2013년	2월	14187건	335명	22255명
99	2013년	3월	17465건	409명	26461명
100	2013년	4월	18031건	380명	27795명
101	2013년	5월	19686건	420명	29676명
102	2013년	6월	18329건	436명	27673명
103	2013년	7월	18441건	444명	28387명
104	2013년	8월	18691건	412명	29190명
105	2013년	9월	17806건	430명	26938명
106	2013년	10월	19797건	499명	29507명
107	2013년	11월	19396건	479명	28986명
108	2013년	12월	17509건	459명	26749명
109	2014년	1월	15905건	389명	24328명
110	2014년	2월	14061건	325명	21501명
111	2014년	3월	18420건	358명	27714명
112	2014년	4월	18443건	341명	27323명
113	2014년	5월	20314건	398명	30684명
...	...	...	...	...	...

114	2014년	6월	18530건	367명	27740명
115	2014년	7월	19649건	407명	29596명
116	2014년	8월	19603건	359명	30633명
117	2014년	9월	19243건	445명	29052명
118	2014년	10월	20760건	470명	31199명
119	2014년	11월	19872건	476명	29525명
120	2014년	12월	18752건	427명	28202명
121	2015년	1월	16902건	381명	25887명
122	2015년	2월	14939건	296명	22999명
123	2015년	3월	18616건	378명	27921명
124	2015년	4월	19386건	354명	29431명
125	2015년	5월	20543건	389명	31264명
126	2015년	6월	18649건	390명	27469명
127	2015년	7월	20278건	359명	30409명
128	2015년	8월	19864건	405명	30618명
129	2015년	9월	20811건	387명	31256명
130	2015년	10월	21587건	470명	32436명
131	2015년	11월	21170건	410명	31773명
132	2015년	12월	19290건	402명	28937명
133	2016년	1월	17067건	336명	26323명
134	2016년	2월	15664건	292명	24021명
135	2016년	3월	17946건	310명	26609명
136	2016년	4월	19452건	353명	29216명
137	2016년	5월	19554건	347명	29172명
138	2016년	6월	17977건	321명	26668명
139	2016년	7월	18955건	358명	28586명
140	2016년	8월	18398건	336명	28017명
141	2016년	9월	17883건	375명	26761명
142	2016년	10월	19918건	440명	29635명
143	2016년	11월	19234건	416명	28520명

144	2016년	12월	18869건	408명	28192명
145	2017년	1월	16970건	353명	26099명
146	2017년	2월	14832건	280명	22323명
147	2017년	3월	17047건	295명	25046명
148	2017년	4월	17717건	293명	26530명
149	2017년	5월	18502건	366명	27268명
150	2017년	6월	18047건	315명	26454명
151	2017년	7월	18158건	357명	27362명
152	2017년	8월	18682건	353명	28162명
153	2017년	9월	19891건	419명	29371명
154	2017년	10월	18863건	420명	28698명
155	2017년	11월	19377건	379명	28472명
156	2017년	12월	18249건	355명	27044명
157	2018년	1월	17026건	304명	25438명
158	2018년	2월	16208건	275명	24630명
159	2018년	3월	17022건	310명	25015명
160	2018년	4월	17992건	303명	26643명
161	2018년	5월	18636건	309명	27834명
162	2018년	6월	18082건	266명	26574명
163	2018년	7월	18699건	315명	28104명
164	2018년	8월	18335건	357명	27749명
165	2018년	9월	18371건	348명	27751명
166	2018년	10월	19738건	373명	28836명
167	2018년	11월	19029건	298명	28000명
168	2018년	12월	18010건	323명	26463명
합계			3084976 건	73436 명	4751813 명

년도 선택시

# Redux 연습문제

2007년도

▼

번호	년도	월	교통 사고 건수	사망자 수	부상자 수
25	2007년	1월	14914건	468명	23975명
26	2007년	2월	14696건	446명	23717명
27	2007년	3월	18166건	476명	28811명
28	2007년	4월	18055건	460명	28555명
29	2007년	5월	19264건	516명	30532명
30	2007년	6월	18310건	538명	28662명
31	2007년	7월	18037건	503명	28676명
32	2007년	8월	17982건	512명	29711명
33	2007년	9월	18506건	532명	29583명
34	2007년	10월	19026건	582명	29802명
35	2007년	11월	17815건	565명	27674명
36	2007년	12월	16891건	568명	26208명
합계			211662 건	6166 명	335906 명