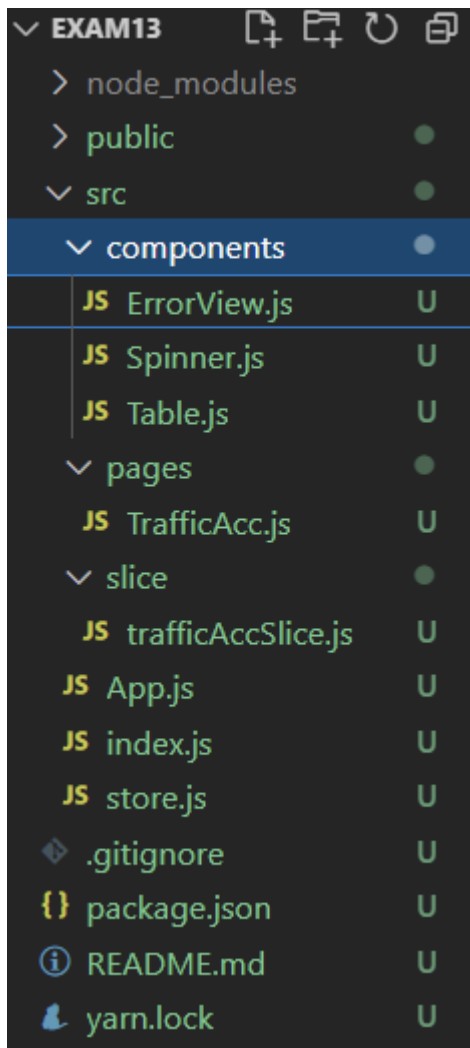


연습문제 13

파일



index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import { BrowserRouter } from 'react-router-dom';

import { Provider } from 'react-redux';
import store from './store';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <Provider store={store}>
      <BrowserRouter>
        <App />
      </BrowserRouter>
    </Provider>
  </React.StrictMode>
);
```

```

    </Provider>
  </React.StrictMode>
);

```

App.js

```

import React from "react";
import TrafficAcc from "../pages/TrafficAcc";

function App() {
  return (
    <div>
      <h1>exam 13</h1>
      <hr/>
      <TrafficAcc/>
    </div>
  );
}

export default App;

```

slice > trafficAccSlice.js

```

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

export const getList = createAsyncThunk("trafficAccSlice/getList", async(payload, {rejectedWithValue}) => {
  let result = null;
  try {
    payload ? result = await axios.get(`http://localhost:3001/traffic_acc?year=${payload}`) : result = await axios.get(`http://localhost:3001/traffic_acc`)
  } catch(err) {
    result = rejectedWithValue(err.reponse)
  }
  return result;
})

const trafficAccSlice = createSlice({
  name: 'trafficAcc',
  initialState: {
    data: null,
    loading: false,
    error: null
  },
  reducers: {},
  extraReducers: {
    [getList.pending]: (state, { payload }) => {

```

```

        return { ...state, loading: true }
      },
      [getList.fulfilled]: (state, { payload }) => {
        return {
          data: payload?.data,
          loading: false,
          error: null
        }
      },
      [getList.rejected]: (state, { payload }) => {
        return {
          data: payload?.data,
          loading: false,
          error: {
            code: payload?.status ? payload.status : 500,
            message: payload?.statusText ? payload.statusText : 'Server
Error'
          }
        }
      }
    }
  }
})

export default trafficAccSlice.reducer

```

pages > TrafficAcc.js

```

import React, { memo } from 'react';
import styled from 'styled-components';
import { useSelector, useDispatch } from 'react-redux';
import { getList } from '../slice/trafficAccSlice';

import Spinner from '../components/Spinner';
import Table from '../components/Table';
import ErrorView from '../components/ErrorView';
/** 드롭다운을 배치하기 위한 박스 */
const SelectContainer = styled.div`
  position: sticky;
  top: 0;
  background-color: #fff;
  border-top: 1px solid #eee;
  border-bottom: 1px solid #eee;
  padding: 10px 0;
  margin: 0;

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

```

```

const TrafficAcc = memo(() => {
  let totalAcc = 0;
  let totalDeath = 0;
  let totalInjury = 0;
  const dispatch = useDispatch();
  const {data, loading, error} = useSelector(state => state.trafficAcc)
  const [trafficData, setTrafficData] = React.useState();
  React.useEffect(() => {
    dispatch(getList(trafficData))
  }, [dispatch, trafficData])
  const onChange = React.useCallback((e) => {
    e.preventDefault();
    const current = e.target;
    const selectValue = current[current.selectedIndex].value
    setTrafficData(selectValue)
  }, [setTrafficData])
  return (
    <div>
      <Spinner visible={loading}/>
      <SelectContainer>
        <select name="year" onChange={onChange}>
          <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>
            <tr>
              <th>번호</th>
              <th>년도</th>
              <th>월</th>
              <th>교통사고 건수</th>
              <th>사망자 수</th>
              <th>부상자 수</th>
            </tr>
          </thead>
          <tbody>
            {data && data.map(({
              id, year, month, accident, death, injury
            }, i) => {
              totalAcc += accident;
              totalDeath += death;
              totalInjury += injury
              return (
                <tr key={id}>
                  <td>{id}</td>
                  <td>{year}년</td>
                  <td>{month}월</td>
                  <td>{accident.toLocaleString('ko-KR')}건</td>

```

```

        <td>{death.toLocaleString('ko-KR')}명</td>
        <td>{injury.toLocaleString('ko-KR')}명</td>
      </tr>
    )
  }
}
</tbody>
<tfoot>
  <tr>
    <th colSpan="3">합계</th>
    <th>{totalAcc.toLocaleString('ko-KR')}건</th>
    <th>{totalDeath.toLocaleString('ko-KR')}명</th>
    <th>{totalInjury.toLocaleString('ko-KR')}명</th>
  </tr>
</tfoot>
</Table>
  )
</div>
);
});

export default TrafficAcc;

```

components > Spinner.js

```

import React from 'react';
import PropTypes from "prop-types"
import styled from 'styled-components';

/** 로딩바 컴포넌트 */
// --> https://mhnpd.github.io/react-loader-spinner/
import {Bars} from 'react-loader-spinner'

/** 로딩바 뒤에 표시될 반투명 막 */
const TransLayer = styled.div`
  position: fixed;
  left: 0;
  right: 0;
  z-index: 9999;
  background-color: #0003;
  width: 100%;
  height: 100%;
`;

const Spinner = ({ visible, color, width, height }) => {
  return (
    <div>
      {visible &&
        <TransLayer>
          <Bars
            color={color}
            height={height}
            width={width}

```

```

        wrapperStyle={{
          position: 'absolute',
          zIndex: 10000,
          left: '50%',
          top: '50%',
          marginLeft: (-width/2)+'px',
          marginTop: (-height/2)+'px'
        }}
      />
    </TransLayer>
  }
</div>
);
};

/**기본값 정의 */
Spinner.defaultProps = {
  visible: false,
  color: '#06f',
  width: 100,
  height: 100
};

/** 데이터 타입 설정 */
Spinner.propTypes = {
  visible: PropTypes.bool.isRequired,
  color: PropTypes.string,
  width: PropTypes.number,
  height: PropTypes.number,
}

export default Spinner;

```

components > Table.js

```

import styled from 'styled-components';

/** 표에 CSS를 적용한 styledComponents */
const Table = styled.table`
  border-collapse: collapse;
  border-top: 3px solid #16B;
  font-size: 14px;
  text-align: center;
  margin: auto;
  width: 100%;

  th {
    color: #16B;
    background: #f0f6f9;
    padding: 10px;
    border: 1px solid #ddd;
  }

```

```

    &:first-child {
      border-left: 0;
    }

    &:last-child {
      border-right: 0;
    }
  }

  td {
    padding: 10px;
    border: 1px solid #ddd;
    &:first-child {
      border-left: 0;
    }
    &:last-child {
      border-right: 0;
    }
  }
}

export default Table;

```

components > ErrorView.js

```

import React, { memo } from 'react';

const ErrorView = memo(({error}) => {
  return (
    <div>
      <h1>Oops~!!! {error.code} Error.</h1>
      <hr/>
      <p>{error.message}</p>
    </div>
  );
});

export default ErrorView;

```

결과

exam 13

-- 년도 선택-- ▾

번호	년도	월	교통사고 건수	사망자 수	부상자 수
1	2005년	1월	15,494건	504명	25,413명
2	2005년	2월	13,244건	431명	21,635명
3	2005년	3월	16,580건	477명	25,550명
4	2005년	4월	17,817건	507명	28,131명
5	2005년	5월	19,085건	571명	29,808명
6	2005년	6월	18,092건	476명	28,594명
7	2005년	7월	18,675건	528명	29,984명
8	2005년	8월	19,035건	562명	31,603명
9	2005년	9월	18,759건	577명	29,831명
10	2005년	10월	19,757건	639명	31,597명
11	2005년	11월	19,129건	574명	30,337명
12	2005년	12월	18,504건	530명	29,750명
13	2006년	1월	14,971건	420명	24,533명
14	2006년	2월	14,270건	373명	22,903명
15	2006년	3월	16,767건	465명	26,013명
16	2006년	4월	17,948건	469명	28,725명
17	2006년	5월	19,140건	531명	30,279명
18	2006년	6월	17,435건	455명	27,032명
19	2006년	7월	18,634건	516명	29,978명
20	2006년	8월	18,794건	585명	30,882명
21	2006년	9월	19,293건	580명	30,186명
22	2006년	10월	19,100건	651명	30,715명
23	2006년	11월	19,877건	701명	31,270명
24	2006년	12월	17,516건	581명	27,713명
25	2007년	1월	14,914건	468명	23,975명
26	2007년	2월	14,696건	446명	23,717명
27	2007년	3월	18,166건	476명	28,811명
28	2007년	4월	18,055건	460명	28,555명
29	2007년	5월	19,264건	516명	30,532명
30	2007년	6월	18,310건	538명	28,662명
31	2007년	7월	18,037건	503명	28,676명
32	2007년	8월	17,982건	512명	29,711명
33	2007년	9월	18,506건	532명	29,583명
34	2007년	10월	19,026건	582명	29,802명
35	2007년	11월	17,815건	565명	27,674명
36	2007년	12월	16,891건	568명	26,208명
37	2008년	1월	14,741건	444명	23,522명
38	2008년	2월	14,176건	428명	23,282명
39	2008년	3월	16,752건	502명	26,284명
40	2008년	4월	17,498건	476명	27,364명
41	2008년	5월	19,515건	493명	30,935명

42	2008년	6월	18,824건	423명	28,872명
43	2008년	7월	19,562건	452명	30,553명
44	2008년	8월	18,545건	529명	29,705명
45	2008년	9월	18,023건	502명	27,894명
46	2008년	10월	19,926건	574명	30,882명
47	2008년	11월	19,597건	528명	30,585명
48	2008년	12월	18,663건	519명	29,084명
49	2009년	1월	16,532건	475명	27,373명
50	2009년	2월	15,502건	405명	24,429명
51	2009년	3월	18,224건	468명	27,967명
52	2009년	4월	19,153건	439명	29,521명
53	2009년	5월	20,237건	519명	31,743명
54	2009년	6월	19,249건	433명	29,084명
55	2009년	7월	20,281건	454명	31,545명
56	2009년	8월	19,993건	514명	32,200명
57	2009년	9월	21,004건	500명	31,700명
58	2009년	10월	21,440건	592명	33,255명
59	2009년	11월	20,156건	522명	31,062명
60	2009년	12월	20,219건	517명	31,996명
61	2010년	1월	17,124건	438명	27,655명
62	2010년	2월	15,803건	406명	24,968명
63	2010년	3월	17,801건	396명	27,350명
64	2010년	4월	18,386건	395명	28,191명
65	2010년	5월	20,372건	447명	31,521명
66	2010년	6월	18,910건	474명	28,311명
67	2010년	7월	18,965건	453명	29,577명
68	2010년	8월	19,383건	410명	31,032명
69	2010년	9월	19,439건	436명	30,470명
70	2010년	10월	21,575건	619명	33,282명
71	2010년	11월	20,271건	541명	30,907명
72	2010년	12월	18,849건	490명	29,194명
73	2011년	1월	16,454건	395명	25,611명
74	2011년	2월	14,208건	339명	22,493명
75	2011년	3월	16,832건	338명	25,371명
76	2011년	4월	18,570건	429명	28,534명
77	2011년	5월	19,414건	393명	29,906명
78	2011년	6월	18,505건	415명	28,469명
79	2011년	7월	19,253건	462명	29,576명
80	2011년	8월	18,896건	472명	30,034명
81	2011년	9월	19,065건	474명	29,055명
82	2011년	10월	20,952건	520명	32,133명
83	2011년	11월	20,283건	514명	30,917명
84	2011년	12월	19,279건	478명	29,292명
85	2012년	1월	16,818건	418명	26,281명
86	2012년	2월	16,656건	393명	25,998명
87	2012년	3월	18,255건	403명	27,899명
88	2012년	4월	19,377건	482명	29,628명

순	연월	일	입원건	퇴원	병상
89	2012년	5월	19,672건	444명	30,163명
90	2012년	6월	18,854건	476명	28,314명
91	2012년	7월	19,333건	416명	29,642명
92	2012년	8월	18,407건	409명	29,051명
93	2012년	9월	19,234건	486명	29,462명
94	2012년	10월	19,557건	533명	29,759명
95	2012년	11월	19,750건	508명	30,054명
96	2012년	12월	17,748건	423명	28,314명
97	2013년	1월	16,016건	389명	25,094명
98	2013년	2월	14,187건	335명	22,255명
99	2013년	3월	17,465건	409명	26,461명
100	2013년	4월	18,031건	380명	27,795명
101	2013년	5월	19,686건	420명	29,676명
102	2013년	6월	18,329건	436명	27,673명
103	2013년	7월	18,441건	444명	28,387명
104	2013년	8월	18,691건	412명	29,190명
105	2013년	9월	17,806건	430명	26,938명
106	2013년	10월	19,797건	499명	29,507명
107	2013년	11월	19,396건	479명	28,986명
108	2013년	12월	17,509건	459명	26,749명
109	2014년	1월	15,905건	389명	24,328명
110	2014년	2월	14,061건	325명	21,501명
111	2014년	3월	18,420건	358명	27,714명
112	2014년	4월	18,443건	341명	27,323명
113	2014년	5월	20,314건	398명	30,684명
114	2014년	6월	18,530건	367명	27,740명
115	2014년	7월	19,649건	407명	29,596명
116	2014년	8월	19,603건	359명	30,633명
117	2014년	9월	19,243건	445명	29,052명
118	2014년	10월	20,760건	470명	31,199명
119	2014년	11월	19,872건	476명	29,525명
120	2014년	12월	18,752건	427명	28,202명
121	2015년	1월	16,902건	381명	25,887명
122	2015년	2월	14,939건	296명	22,999명
123	2015년	3월	18,616건	378명	27,921명
124	2015년	4월	19,386건	354명	29,431명
125	2015년	5월	20,543건	389명	31,264명
126	2015년	6월	18,649건	390명	27,469명
127	2015년	7월	20,278건	359명	30,409명
128	2015년	8월	19,864건	405명	30,618명
129	2015년	9월	20,811건	387명	31,256명
130	2015년	10월	21,587건	470명	32,436명
131	2015년	11월	21,170건	410명	31,773명
132	2015년	12월	19,290건	402명	28,937명
133	2016년	1월	17,067건	336명	26,323명
134	2016년	2월	15,664건	292명	24,021명

135	2016년	3월	17,946건	310명	26,609명
136	2016년	4월	19,452건	353명	29,216명
137	2016년	5월	19,554건	347명	29,172명
138	2016년	6월	17,977건	321명	26,668명
139	2016년	7월	18,955건	358명	28,586명
140	2016년	8월	18,398건	336명	28,017명
141	2016년	9월	17,883건	375명	26,761명
142	2016년	10월	19,918건	440명	29,635명
143	2016년	11월	19,234건	416명	28,520명
144	2016년	12월	18,869건	408명	28,192명
145	2017년	1월	16,970건	353명	26,099명
146	2017년	2월	14,832건	280명	22,323명
147	2017년	3월	17,047건	295명	25,046명
148	2017년	4월	17,717건	293명	26,530명
149	2017년	5월	18,502건	366명	27,268명
150	2017년	6월	18,047건	315명	26,454명
151	2017년	7월	18,158건	357명	27,362명
152	2017년	8월	18,682건	353명	28,162명
153	2017년	9월	19,891건	419명	29,371명
154	2017년	10월	18,863건	420명	28,698명
155	2017년	11월	19,377건	379명	28,472명
156	2017년	12월	18,249건	355명	27,044명
157	2018년	1월	17,026건	304명	25,438명
158	2018년	2월	16,208건	275명	24,630명
159	2018년	3월	17,022건	310명	25,015명
160	2018년	4월	17,992건	303명	26,643명
161	2018년	5월	18,636건	309명	27,834명
162	2018년	6월	18,082건	266명	26,574명
163	2018년	7월	18,699건	315명	28,104명
164	2018년	8월	18,335건	357명	27,749명
165	2018년	9월	18,371건	348명	27,751명
166	2018년	10월	19,738건	373명	28,836명
167	2018년	11월	19,029건	298명	28,000명
168	2018년	12월	18,010건	323명	26,463명
합계			3,084,976건	73,436명	4,751,813명

exam 13

2013년도 ▾

번호	년도	월	교통사고 건수	사망자 수	부상자 수
97	2013년	1월	16,016건	389명	25,094명
98	2013년	2월	14,187건	335명	22,255명
99	2013년	3월	17,465건	409명	26,461명
100	2013년	4월	18,031건	380명	27,795명
101	2013년	5월	19,686건	420명	29,676명

102	2013년	6월	18,329건	436명	27,673명
103	2013년	7월	18,441건	444명	28,387명
104	2013년	8월	18,691건	412명	29,190명
105	2013년	9월	17,806건	430명	26,938명
106	2013년	10월	19,797건	499명	29,507명
107	2013년	11월	19,396건	479명	28,986명
108	2013년	12월	17,509건	459명	26,749명
합계			215,354건	5,092명	328,711명