

# EXAM 10 박정모

2022-05-17

## 실행결과 스크린샷

React App

localhost:3000

--년도 선택--

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

React App

localhost:3000

2007년도

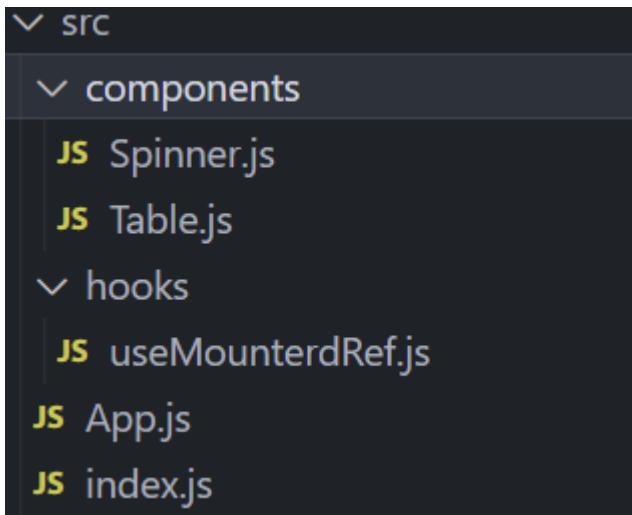
번호	년도	월	교통사고 건수	사망자 수	부상자 수
25	2007년	1월	14,914건	468명	23,975명
26	2007년	2월	14,696건	446명	23,717명
27	2007년	3월	18,166건	476명	28,811명
28	2007년	4월	18,055건	460명	28,555명
29	2007년	5월	19,264건	516명	30,532명
30	2007년	6월	18,310건	538명	28,662명
31	2007년	7월	18,037건	503명	28,676명
32	2007년	8월	17,982건	512명	29,711명
33	2007년	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
합계			211,662건	6,166명	335,906명

### EXAM 10

2007년도

번호	년도	월	교통사고 건수	사망자 수	부상자 수
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
합계			211,662건	6,166명	335,906명

## 파일 구조



## App.js

```
import React from "react";
import useAxios from "axios-hooks";
import useMountedRef from "../hooks/useMounterdRef";
import Table from "../components/Table";
import Spinner from "../components/Spinner";
import styled from "styled-components";

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

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

const App = () => {
  // 선택된 select 값을 저장할 상태값
  const [state, setState] = React.useState("");

  // 마운트됨과 동시에 axios get
  const [{ data, loading, error }, refetch] =
    useAxios("http://localhost:4000/traffic_acc");

  // 드롭다운이 선택되었을 때 그 값을 state에 적용하는 함수
  const onSelectChange = React.useCallback((e) => {
```

```

    e.preventDefault();
    const current = e.target;

    const newState = current[current.selectedIndex].value;

    console.log(newState);
    setState(newState);
  }, []);

  const mountedRef = useMountedRef();

  // 사고건수, 사망자, 부상자의 합을 저장할 변수
  let accSum = 0;
  let deaSum = 0;
  let injSum = 0;

  // state값이 바뀌었을 때, 바뀐 값을 params로 전달하여 fetch함수를 실행
  React.useEffect(() => {
    if (mountedRef.current) {
      // refetch에 params로 전달할 변수
      const params = {};
      // -- 년도 선택 -- 을 골랐을 때는 params를 전달하지 않음
      if (state) {
        params["year"] = state;
      }
      refetch({
        params: params,
      });
    }
  }, [state, mountedRef, refetch]);

  // Ajax 통신이 실패했을 때 실행할 조건문
  if (error) {
    console.error(error);
    return (
      <div>
        <h1>Oops~! {error.code} Error!</h1>
        <hr />
        <p>{error.message}</p>
      </div>
    );
  }

  return (
    <div>
      <h1>EXAM 10</h1>
      <ht />

      <Spinner visible={loading} />
      <SelectContainer>
        <select name="year" onChange={onSelectChange}>
          <option value="">-- 년도 선택 --</option>
          <option value="2005">2005년도</option>
          <option value="2006">2006년도</option>
        </select>
      </SelectContainer>
    </div>
  );
}

```

```

        <option value="2007">2007년도</option>
        <option value="2008">2008년도</option>
        <option value="2009">2009년도</option>
        <option value="2010">2010년도</option>
        <option value="2011">2011년도</option>
        <option value="2012">2012년도</option>
        <option value="2013">2013년도</option>
        <option value="2014">2014년도</option>
        <option value="2015">2015년도</option>
        <option value="2016">2016년도</option>
        <option value="2017">2017년도</option>
        <option value="2018">2018년도</option>
    </select>
</SelectContainer>
<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)
=> {
                accSum += accident;
                deaSum += death;
                injSum += 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>{accSum.toLocaleString('ko-KR')}건</th>
            <th>{deaSum.toLocaleString('ko-KR')}명</th>
            <th>{injSum.toLocaleString('ko-KR')}명</th>
        </tr>
    </tfoot>
</Table>
</div>

```

```
    );  
};  
  
export default App;
```