

[illegible]

13-EXAM

2019년도 ▾

번호	년도	월	교통사고	사망자	부상자
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	27460명
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
합계			232035건	4621명	350400명

ActionsSettings

📌

Reset

Revert

Sweep

Commit

ReactApp

filter...

@@INIT6:25:00.24

Traffic/getTraffic/pending+00:00.40

Traffic/getTraffic/pending+00:00.01

Traffic/getTraffic/fulf...+00:00.17

Traffic/getTraffic/fulf...+00:00.67

Traffic/getTraffic/pending+01:11.75

Traffic/getTraffic/fulf...+00:00.16

Action

TreeChartRaw

type (pin): "Traffic/getTraffic/fulfilled"

▼ payload (pin)

data (pin)

▶ 0 (pin): { id: 121, year: 2015, month: 1, ... }

▶ 1 (pin): { id: 122, year: 2015, month: 2, ... }

▶ 2 (pin): { id: 123, year: 2015, month: 3, ... }

▶ 3 (pin): { id: 124, year: 2015, month: 4, ... }

▶ 4 (pin): { id: 125, year: 2015, month: 5, ... }

▶ 5 (pin): { id: 126, year: 2015, month: 6, ... }

▶ 6 (pin): { id: 127, year: 2015, month: 7, ... }

▶ 7 (pin): { id: 128, year: 2015, month: 8, ... }

▶ 8 (pin): { id: 129, year: 2015, month: 9, ... }

▶ 9 (pin): { id: 130, year: 2015, month: 10, ... }

▶ 10 (pin): { id: 131, year: 2015, month: 11, ... }

▶ 11 (pin): { id: 132, year: 2015, month: 12, ... }

status (pin): 200

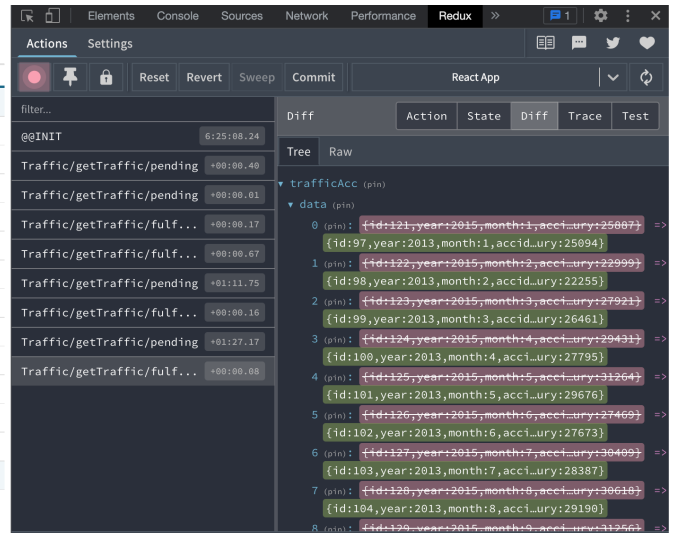
statusText (pin): "OK"

Traffic/getTraffic/fulfilled (6)

Live1x2x

13-EXAM

번호	년도	월	교통사고	사망자	부상자
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
합계			215354건	5092명	328711명



예제와 동일한 코드는 생략하였습니다

src/slices/TrafficSlice.js

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

export const getTraffic = createAsyncThunk(
  'Traffic/getTraffic',
  async (payload, { rejectWithValue }) => {
    let result = null;
    try {
      result = await axios.get('http://localhost:3001/traffic_acc', {
        params: {
          year: payload,
        },
      });
    } catch (err) {
      result = rejectWithValue(err.response);
    }
    return result;
  }
);

const Traffic = createSlice({
  name: 'trafficAcc',
  initialState: {
    data: null,
    loading: false,
    error: null,
  },
  reducers: {},
  extraReducers: {
    [getTraffic.pending]: (state, { payload }) => {
      return { ...state, loading: true };
    },
  },
});
```

```

[getTraffic.fulfilled]: (state, { payload }) => {
  return {
    data: payload?.data,
    loading: false,
    error: null,
  };
},

[getTraffic.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 Traffic.reducer;

```

src/pages/Traffic.js

```

import React, { memo } from 'react';

import { useSelector, useDispatch } from 'react-redux';
import { getTraffic } from '../slices/TrafficSlice';
import Spinner from '../components/Spinner';
import Table from '../components/Table';
import ErrorView from '../components/ErrorView';

const Traffic = memo(() => {
  const dispatch = useDispatch();
  const { data, loading, error } = useSelector((state) =>
state.trafficAcc);
  const [targetDt, setTargetDt] = React.useState('');
  React.useEffect(() => {
    if (targetDt > 0) {
      dispatch(getTraffic(targetDt));
    } else {
      dispatch(getTraffic());
    }
  }, [dispatch, targetDt]);
  const onSelectChange = React.useCallback((e) => {
    e.preventDefault();
    setTargetDt(e.target.value);
  }, []);

```

```

let sum1 = 0;
let sum2 = 0;
let sum3 = 0;

const rendering = () => {
  const result = [];
  for (let i = 2005; i <= 2018; i++) {
    result.push(
      <option key={i} value={i}>
        {i}년도
      </option>
    );
  }
  return result;
};

return (
  <div>
    <Spinner visible={loading} />

    <select name="year" onChange={onSelectChange}>
      <option value="">--년도 선택--</option>
      {rendering()}
    </select>

    {error ? (
      <ErrorView error={error} />
    ) : (
      <Table>
        <thead>
          <tr>
            <th>번호</th>
            <th>년도</th>
            <th>월</th>
            <th>교통사고</th>
            <th>사망자</th>
            <th>부상자</th>
          </tr>
        </thead>
        <tbody>
          {data &&
            data.map(({ id, year, month, accident, death, injury }, i)
=> {
              sum1 += accident;
              sum2 += death;
              sum3 += injury;
              return (
                <tr key={i}>
                  <td>{id}</td>
                  <td>{year}년</td>
                  <td>{month}월</td>
                  <td>{accident}건</td>
                  <td>{death}명</td>
                  <td>{injury}명</td>

```

```
        </tr>
      );
    })}
  </tbody>
  <tfoot>
    <tr>
      <th colSpan="3">합계</th>
      <th>{sum1}건</th>
      <th>{sum2}명</th>
      <th>{sum3}명</th>
    </tr>
  </tfoot>
</Table>
  })
</div>
);
});
export default Traffic;
```