

한주애 과제

2022-05-24

store.js

```
import {configureStore} from '@reduxjs/toolkit';
import TrafficSlice from './slices/TrafficSlice';

const store = configureStore({
  reducer: {
    traffic : TrafficSlice
  },
  middleware: (getDefaultMiddleware) => getDefaultMiddleware({serializableCheck:
false}),
  devTools: true
});

export default store;
```

TrafficSlice.js

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

export const getTrafficList = createAsyncThunk("TrafficSlice/getTrafficList",
async (payload, {rejectWithValue}) => {
  let result = null;

  try{
    result = await axios.get('http://localhost:3001/traffic_acc', {
      params: {
        year: payload.year
      }
    });
  }

  if(result.data.faultInfo !== undefined){
    const err = new Error();
    err.response = {status: 500, statusText:
result.data.faultInfo.message};
    throw err;
  }
} catch (err){
  result = rejectWithValue(err.response);
}
return result;
});
```

```

const TrafficSlice = createSlice({
  name: 'traffic',
  initialState: {
    data: null,
    loading: false,
    error: null
  },
  reducers: {},
  extraReducers: {
    [getTrafficList.pending] : (state, {payload}) => {
      return {...state, loading: true}
    },
    [getTrafficList.fulfilled] : (state, {payload}) => {
      return {
        data: payload?.data,
        loading: false,
        error: null
      }
    },
    [getTrafficList.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 TrafficSlice.reducer;

```

app.js

```

import React from 'react';
import Traffic from './pages/Traffic';

const App = () => {
  return (
    <div>
      <h1>Exam13</h1>
      <hr />
      <Traffic />
    </div>
  );
};

export default App;

```

Traffic.js

```
import React from 'react';
import Spinner from '../components/Spinner';
import Table from '../components/Table';
import ErrorView from '../components/ErrorView';
import styled from 'styled-components';

import {useSelector, useDispatch} from 'react-redux';
import {getTrafficList} from '../slices/TrafficSlice';
import useMountedRef from '../hooks/useMountedRef';

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 Traffic = () => {
  const dispatch = useDispatch();
  const {data, loading, error} = useSelector((state) => state.traffic)
  const [year, setYear] = React.useState();

  React.useEffect(() => {
    dispatch(getTrafficList({year: year}))
  }, [dispatch, year]);

  const onSelectChange = React.useCallback(e => {
    e.preventDefault();
    const current = e.target;
    const value = current[current.selectedIndex].value;
    setYear(value);
  }, [])

  return (
    <div>
      <Spinner visible={loading} />
      {error ? (<ErrorView error={error} />) : (
        <>
          <SelectContainer>

```

```

        <select name="year" onChange={onSelectChange}>
          <option value="">---년도---</option>
          {[...new Array(2018-2005+1)].map((v, i) => {
            return (<option key={i} value={2005+i}>{2005+i}년
도</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) => {
            return(
              <tr key={id}>
                <td>{id}</td>
                <td>{year}년</td>
                <td>{month}월</td>
                <td>{accident.toLocaleString()}건</td>
                <td>{death.toLocaleString()}명</td>
                <td>{injury.toLocaleString()}명</td>
              </tr>
            );
          })}
      </tbody>
      <tfoot>
        <tr>
          <th colspan="3">합계</th>
          <th>{data && data.map((v, i) => v.accident).reduce((p,
c) => p+c, 0).toLocaleString()}건</th>
          <th>{data && data.map((v, i) => v.death).reduce((p, c)
=> p+c, 0).toLocaleString()}명</th>
          <th>{data && data.map((v, i) => v.injury).reduce((p,
c) => p+c, 0).toLocaleString()}명</th>
        </tr>
      </tfoot>
    </Table>
  </>
</div>
);
};

export default React.memo(Traffic);

```

Exam13

---년도--- ▾

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

Exam13

2009년도 ▾

번호	년도	월	교통사고 건수	사망자 수	부상자 수
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
합계			231,990건	5,838명	361,875명