

유다현 과제

2022-05-24

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>
);
```

App.js

```
import React, { memo } from 'react';
import {Routes,Route} from "react-router-dom";
import MovieRank from "./pages/MovieRank";
import MenuLink from "./components/MenuLink";
import Car from "./pages/Car";

const App = memo(() => {
  return (
    <div>
      <h2>영화순위</h2>
      <MenuLink to="/movie">영화</MenuLink>
      <MenuLink to="/car">교통사고 현황</MenuLink>
      <Routes>
        <Route path="/car" element={<Car/>}></Route>
        <Route path="/movie" element={<MovieRank/>}></Route>
      </Routes>
    </div>
  );
});
```

```

    </div>
  );
});

export default App;

```

Car.js

```

import React, { memo } from 'react';
import { getTraffic } from "../slice/CarSlice";
import Table from "../components/Table";
import Spinner from "../components/Spinner";

import { useSelector, useDispatch } from "react-redux";

const car = memo(() => {
  const { data, loading, error } = useSelector((state) => state.car);

  const [targetDt, settargetDt] = React.useState("");
  const dispatch = useDispatch();

  React.useEffect(() => {
    dispatch(getTraffic(targetDt ? { params: { year: targetDt } } : null))
  }, [dispatch, targetDt])

  const onSelectChange = React.useCallback((e) => {
    e.preventDefault();
    settargetDt(parseInt(e.target.value));
    console.log(parseInt(e.target.value))
  }, []);

  return (
    <div>
      <Spinner loading={loading}/>
      <select onChange={onSelectChange}>
        <option>--년도선택</option>
        {[...new Array(2018 - 2005 + 1)].map((v, i) => {
          return (<option key={i} value={2005 + i}>{2005 +
i} 년도
          </option>);
        })}
      </select>
      {error ? (
        <>
          <p>에러입니다.</p>
        </>
      ) : (
        <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((v,i)=>{
                return(
                    <tr key={i}>
                        <td>{v.id}</td>
                        <td>{v.year}</td>
                        <td>{v.month}</td>
                        <td>{v.accident.toLocaleString()}건
                    <td>{v.death.toLocaleString()}명
                    <td>{v.injury.toLocaleString()}명
                </tr>
            )
        })
    </tbody>
</Table>
    </div>
    );
});

export default car;

```

CarSlice.js

```

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

//비동기 처리 함수 구현//
//payload는 이 함수를 호출 할 때 전달되는 파라미터
export const getTraffic = createAsyncThunk("CarSlice/getTraffic",async
(payload,{rejectWithValue})=>{
    let result = null;
    try{

```

```

        result = await
    axios.get("http://localhost:3001/traffic_acc",payload);
    }
    catch(e){
        result = rejectWithValue(e.reponse);
    }
    return result;
})

const CarSlice= createSlice({
    name: "car",
    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.extraReducers:"Server Error"
                }
            }
        }
    }
});

export default CarSlice.reducer;

```


React App

localhost:3000/car

영화순위

영화 | [교통사고 현황](#) |

2012 년도 ▾

번호	년도	월	교통사고 건수	사망자 수	부상자 수
85	2012	1	16,818건	418명	26,281명
86	2012	2	16,656건	393명	25,998명
87	2012	3	18,255건	403명	27,899명
88	2012	4	19,372건	483명	29,628명
89	2012	5	19,672건	444명	30,163명
90	2012	6	18,854건	476명	28,314명
91	2012	7	19,333건	416명	29,642명
92	2012	8	18,407건	409명	29,051명
93	2012	9	19,234건	486명	29,462명
94	2012	10	19,557건	533명	29,759명
95	2012	11	19,750건	508명	30,054명
96	2012	12	17,748건	423명	28,314명

React App

localhost:3000/car

영화순위

영화 | [교통사고 현황](#) |

2005 년도 ▾

번호	년도	월	교통사고 건수	사망자 수	부상자 수
1	2005	1	15,494건	504명	25,413명
2	2005	2	13,244건	431명	21,635명
3	2005	3	16,580건	477명	25,550명
4	2005	4	17,817건	507명	28,131명

5	2005	5	19,085건	571명	29,808명
6	2005	6	18,092건	476명	28,594명
7	2005	7	18,675건	528명	29,984명
8	2005	8	19,035건	562명	31,603명
9	2005	9	18,759건	577명	29,831명
10	2005	10	19,757건	639명	31,597명
11	2005	11	19,129건	574명	30,337명
12	2005	12	18,504건	530명	29,750명