

박찬우 리액트 연습문제12

2022-05-24

– src/index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';

// 리덕스 구성을 위한 참조
import { Provider } from 'react-redux';
import Store from './Store';

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

– src/App.jsx

```
import React from 'react';
import TrafficAcc from './pages/TrafficAcc';

const App = () => {
  return (
    <div>
      <h1>Exam12</h1>
      <TrafficAcc />
    </div>
  );
};

export default App;
```

– src/slice/TrafficAccSlice.jsx

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

// TODO: 비동기 처리 함수 구현
```

```
export const getTrafficAcc = createAsyncThunk('traffic/getTrafficAcc',
  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;
  });

// TODO: Slice 정의
const TrafficAccSlice = createSlice({
  name: 'trafficAcc',
  initialState: {
    data: null,
    loading: false,
    error: null,
  },

  extraReducers: {
    [getTrafficAcc.pending]: (state, {payload}) => {
      return { ...state, loading: true }
    },

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

    [getTrafficAcc.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;
```

- src/Store.jsx

```
import { configureStore } from "@reduxjs/toolkit";

import TrafficAccSlice from "../slice/TrafficAccSlice";

const Store = configureStore({
  reducer: {
    trafficAcc: TrafficAccSlice,
  },

  middleware: (getDefaultMiddleware) =>
    getDefaultMiddleware({serializableCheck: false}),
  devTools: true,
});

export default Store;
```

- src/pages/TrafficAcc.jsx

```
import React, { memo, useCallback, useEffect, useState } from 'react';
import { useSelector, useDispatch } from 'react-redux';
import styled from 'styled-components';
import useMountedRef from '../hook/useMounterRef';

import Spinner from '../components/Spinner';
import Table from '../components/Table';
import ErrorPage from '../components/ErrorPage';
import { getTrafficAcc } from '../slice/TrafficAccSlice';

// 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-left: 15px;
  font-size: 16px;
  padding: 5px 10px;
}
`;

const TrafficAcc = () => {

  // hook을 통해 slice가 관리하는 상태값 가져오기
  const { data, loading, error } = useSelector((state) =>
state.trafficAcc);

  // 드롭다운 상태값 정의
  const [getYear, setGetYear] = useState('');

  // dispatch 함수 생성
  const dispatch = useDispatch();

  // 이 컴포넌트가 화면에 마운트 되었는지를 확인하기 위한 hook
  const mountedRef = useMountedRef();

  // 컴포넌트가 마운트 직후 모든 정보 호출
  useEffect(() => {
    dispatch(getTrafficAcc(getYear));
  }, [dispatch, getYear]);

  // 년도 선택 시 ->
  useEffect(() => {
    if(mountedRef.current) {
      dispatch(getTrafficAcc({ year: getYear }));
    }
  }, [mountedRef, dispatch, getYear]);

  const onSelectChange = useCallback(e => {
    e.preventDefault();

    const current = e.target;
    const value = current[current.selectedIndex].value;

    setGetYear(value);
  }, []);

  const head = ['번호', '년도', '월', '교통사고 건수', '사망자 수', '부상자 수'];

  // TODO: 메인화면 구성
  return (
    <div>
      <Spinner visible={loading} />

      <SelectContainer>
        <select name="year" onChange={onSelectChange}>
          <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 ? <ErrorPage error={error} /> : (
    <>
        {data && (
            <Table>
                <thead>
                    <tr>
                        {head.map((v,i) => <th key={i}>{v}</th>)}
                    </tr>
                </thead>
                <tbody>
                    {data && data.map(({ id, year, month, accident, death,
injury }, i) => {
                        return (
                            <tr key={i}>
                                <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.map((v,i) => v.accident).reduce((a,b) => a +
b, 0).toLocaleString()} 건</th>
                        <th>{data.map((v,i) => v.death).reduce((a,b) => a + b,
0).toLocaleString()} 명</th>
                        <th>{data.map((v,i) => v.injury).reduce((a,b) => a + b,
0).toLocaleString()} 명</th>
                    </tr>
                </tfoot>
            </Table>
        )}
    </>
)}
</div>
);
}

export default memo(TrafficAcc);

```

실행결과

Exam12

--년도 선택--

| 번호 | 년도 | 월 | 교통사고 건수 | 사망자 수 | 부상자 수 |
|----|--------|------|----------|-------|----------|
| 1 | 2005 년 | 1 월 | 15,494 건 | 504 명 | 25,413 명 |
| 2 | 2005 년 | 2 월 | 13,244 건 | 431 명 | 21,635 명 |
| 3 | 2005 년 | 3 월 | 16,580 건 | 477 명 | 25,550 명 |
| 4 | 2005 년 | 4 월 | 17,817 건 | 507 명 | 28,131 명 |
| 5 | 2005 년 | 5 월 | 19,085 건 | 571 명 | 29,808 명 |
| 6 | 2005 년 | 6 월 | 18,092 건 | 476 명 | 28,594 명 |
| 7 | 2005 년 | 7 월 | 18,675 건 | 528 명 | 29,984 명 |
| 8 | 2005 년 | 8 월 | 19,035 건 | 562 명 | 31,603 명 |
| 9 | 2005 년 | 9 월 | 18,759 건 | 577 명 | 29,831 명 |
| 10 | 2005 년 | 10 월 | 19,757 건 | 639 명 | 31,597 명 |
| 11 | 2005 년 | 11 월 | 19,129 건 | 574 명 | 30,337 명 |
| 12 | 2005 년 | 12 월 | 18,504 건 | 530 명 | 29,750 명 |
| 13 | 2006 년 | 1 월 | 14,971 건 | 420 명 | 24,533 명 |
| 14 | 2006 년 | 2 월 | 14,270 건 | 373 명 | 22,903 명 |
| 15 | 2006 년 | 3 월 | 16,767 건 | 465 명 | 26,013 명 |
| 16 | 2006 년 | 4 월 | 17,948 건 | 469 명 | 28,725 명 |
| 17 | 2006 년 | 5 월 | 19,140 건 | 531 명 | 30,279 명 |
| 18 | 2006 년 | 6 월 | 17,435 건 | 455 명 | 27,032 명 |
| 19 | 2006 년 | 7 월 | 18,634 건 | 516 명 | 29,978 명 |
| 20 | 2006 년 | 8 월 | 18,794 건 | 585 명 | 30,882 명 |
| 21 | 2006 년 | 9 월 | 19,293 건 | 580 명 | 30,186 명 |
| 22 | 2006 년 | 10 월 | 19,100 건 | 651 명 | 30,715 명 |
| 23 | 2006 년 | 11 월 | 19,877 건 | 701 명 | 31,270 명 |
| 24 | 2006 년 | 12 월 | 17,516 건 | 581 명 | 27,713 명 |
| 25 | 2007 년 | 1 월 | 14,914 건 | 468 명 | 23,975 명 |
| 26 | 2007 년 | 2 월 | 14,696 건 | 446 명 | 23,717 명 |
| 27 | 2007 년 | 3 월 | 18,166 건 | 476 명 | 28,811 명 |
| 28 | 2007 년 | 4 월 | 18,055 건 | 460 명 | 28,555 명 |
| 29 | 2007 년 | 5 월 | 19,264 건 | 516 명 | 30,532 명 |
| 30 | 2007 년 | 6 월 | 18,310 건 | 538 명 | 28,662 명 |
| 31 | 2007 년 | 7 월 | 18,037 건 | 503 명 | 28,676 명 |

Exam12

2008 년도

| 번호 | 년도 | 월 | 교통사고 건수 | 사망자 수 | 부상자 수 |
|----|--------|------|-----------|---------|-----------|
| 37 | 2008 년 | 1 월 | 14,741 건 | 444 명 | 23,522 명 |
| 38 | 2008 년 | 2 월 | 14,176 건 | 428 명 | 23,282 명 |
| 39 | 2008 년 | 3 월 | 16,752 건 | 502 명 | 26,284 명 |
| 40 | 2008 년 | 4 월 | 17,498 건 | 476 명 | 27,364 명 |
| 41 | 2008 년 | 5 월 | 19,515 건 | 493 명 | 30,935 명 |
| 42 | 2008 년 | 6 월 | 18,824 건 | 423 명 | 28,872 명 |
| 43 | 2008 년 | 7 월 | 19,562 건 | 452 명 | 30,553 명 |
| 44 | 2008 년 | 8 월 | 18,545 건 | 529 명 | 29,705 명 |
| 45 | 2008 년 | 9 월 | 18,023 건 | 502 명 | 27,894 명 |
| 46 | 2008 년 | 10 월 | 19,926 건 | 574 명 | 30,882 명 |
| 47 | 2008 년 | 11 월 | 19,597 건 | 528 명 | 30,585 명 |
| 48 | 2008 년 | 12 월 | 18,663 건 | 519 명 | 29,084 명 |
| 합계 | | | 215,822 건 | 5,870 명 | 338,962 명 |

Exam12

2014 년도 ▾

| 번호 | 년도 | 월 | 교통사고 건수 | 사망자 수 | 부상자 수 |
|-----|--------|------|-----------|---------|-----------|
| 109 | 2014 년 | 1 월 | 15,905 건 | 389 명 | 24,328 명 |
| 110 | 2014 년 | 2 월 | 14,061 건 | 325 명 | 21,501 명 |
| 111 | 2014 년 | 3 월 | 18,420 건 | 358 명 | 27,714 명 |
| 112 | 2014 년 | 4 월 | 18,443 건 | 341 명 | 27,323 명 |
| 113 | 2014 년 | 5 월 | 20,314 건 | 398 명 | 30,684 명 |
| 114 | 2014 년 | 6 월 | 18,530 건 | 367 명 | 27,740 명 |
| 115 | 2014 년 | 7 월 | 19,649 건 | 407 명 | 29,596 명 |
| 116 | 2014 년 | 8 월 | 19,603 건 | 359 명 | 30,633 명 |
| 117 | 2014 년 | 9 월 | 19,243 건 | 445 명 | 29,052 명 |
| 118 | 2014 년 | 10 월 | 20,760 건 | 470 명 | 31,199 명 |
| 119 | 2014 년 | 11 월 | 19,872 건 | 476 명 | 29,525 명 |
| 120 | 2014 년 | 12 월 | 18,752 건 | 427 명 | 28,202 명 |
| 합계 | | | 223,552 건 | 4,762 명 | 337,497 명 |

Exam12

2018 년도 ▾

| 번호 | 년도 | 월 | 교통사고 건수 | 사망자 수 | 부상자 수 |
|-----|--------|------|-----------|---------|-----------|
| 157 | 2018 년 | 1 월 | 17,026 건 | 304 명 | 25,438 명 |
| 158 | 2018 년 | 2 월 | 16,208 건 | 275 명 | 24,630 명 |
| 159 | 2018 년 | 3 월 | 17,022 건 | 310 명 | 25,015 명 |
| 160 | 2018 년 | 4 월 | 17,992 건 | 303 명 | 26,643 명 |
| 161 | 2018 년 | 5 월 | 18,636 건 | 309 명 | 27,834 명 |
| 162 | 2018 년 | 6 월 | 18,082 건 | 266 명 | 26,574 명 |
| 163 | 2018 년 | 7 월 | 18,699 건 | 315 명 | 28,104 명 |
| 164 | 2018 년 | 8 월 | 18,335 건 | 357 명 | 27,749 명 |
| 165 | 2018 년 | 9 월 | 18,371 건 | 348 명 | 27,751 명 |
| 166 | 2018 년 | 10 월 | 19,738 건 | 373 명 | 28,836 명 |
| 167 | 2018 년 | 11 월 | 19,029 건 | 298 명 | 28,000 명 |
| 168 | 2018 년 | 12 월 | 18,010 건 | 323 명 | 26,463 명 |
| 합계 | | | 217,148 건 | 3,781 명 | 323,037 명 |