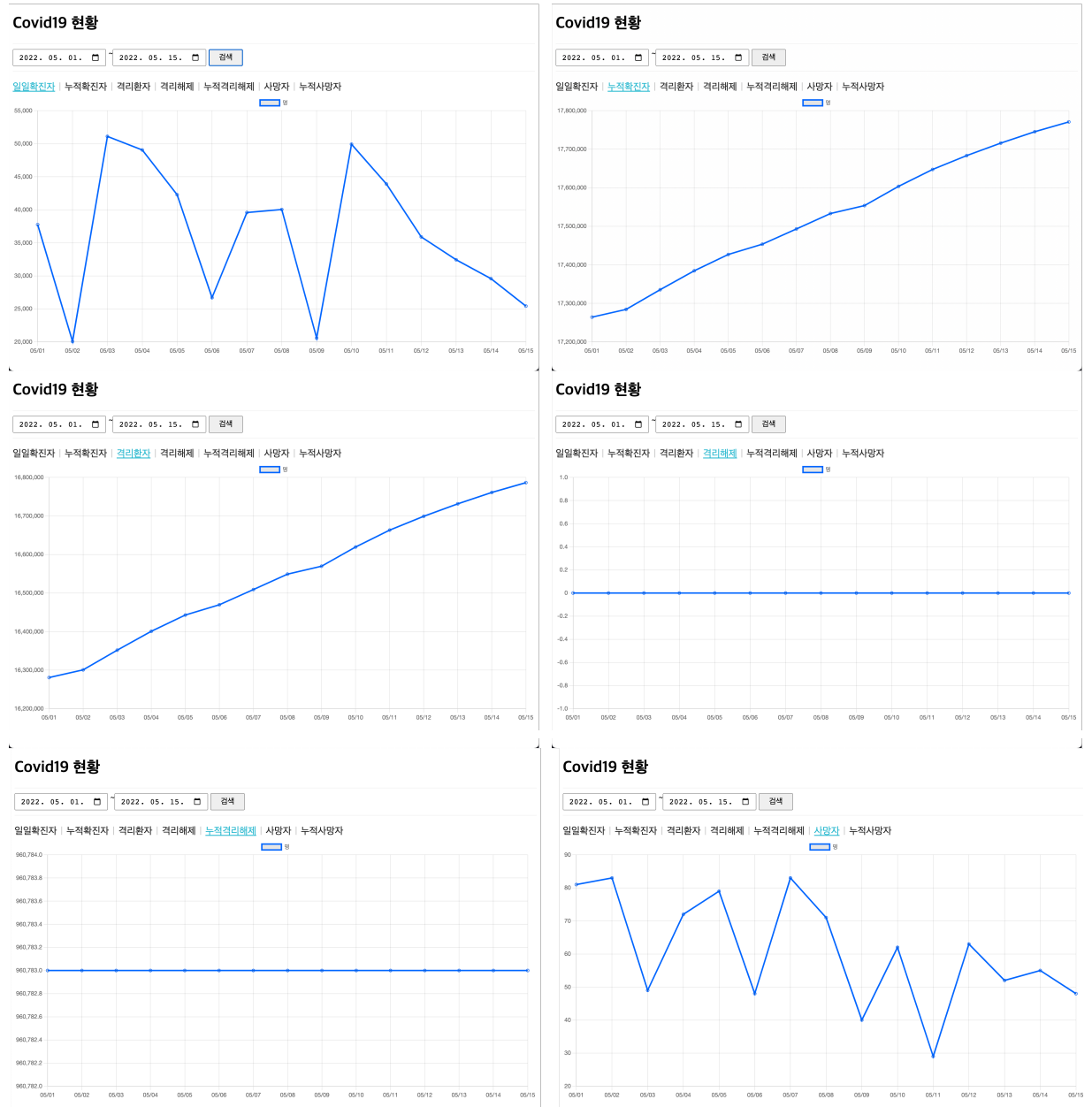


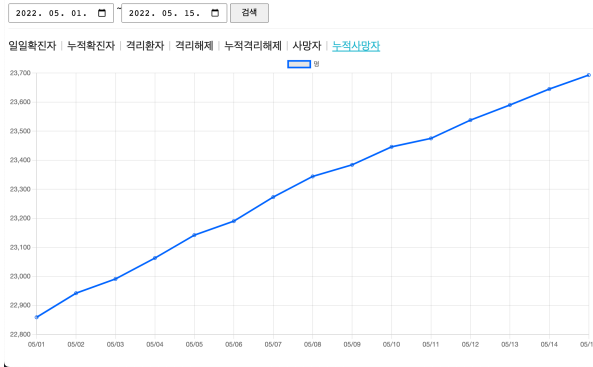
4. JSON 데이터는 다음과 같은 필드를 갖는다.

필드명	설명
id	일련번호
date	날짜(ISO 형식)
active	격리중
confirmed	당일 확진자 수
confirmed_acc	누적 확진자 수
released	당일 격리해제
released_acc	누적 격리해제
death	당일 사망자
death_acc	누적 사망자

5. 검색 시작일과 검색 종료일이 모두 지정된 상태에서 검색 버튼이 눌리지면 pages/covid19.js 파일에 의해 아래와 같이 메뉴가 표시되면서 해당 메뉴에 대한 그래프가 표현된다.



Covid19 현황



<과제 작성시 지시사항>

- (1) 모든 작업은 작업 과정에 대한 소스코드와 중간 출력 결과가 제시된 PowerPoint 보고서 형태로 제출되어야 합니다. (형식에 맞지 않을 경우 0점 처리)
- (2) 제출해야 하는 소스코드는 실행결과가 포함되어야 합니다. 실행결과가 없거나 에러가 나는 상태로 제출할 경우 0점 처리 됩니다.
- (3) 인터넷을 이용한 조사는 가능하나 자신만의 표현으로 재서술해야 인정되며, 표절율이 50%이상이면 0점 처리합니다.
- (4) 타 학생의 과제물과 표현이 일치할 경우 제공자 및 참조자 모두 감점 처리됩니다.
- (5) PowerPoint로 작성된 파일을 PDF로 변환하여 온라인으로 제출하며 본 평가지에는 이름과 서명을 날인하여 별도로 제출합니다. (PPT나 소스파일 원본을 제출하지 마세요.)
- (6) 시험 문제를 해결하면서 겪은 문제점과 본인의 소감을 반드시 포함해야 합니다.
- (7) 모든 문제의 풀이 과정을 하나의 보고서 형태로 병합하여 단일 파일로 제출해야 합니다.

【평가항목 및 배점】

평가항목	평가내용	배점기준	배점	점수
지식/기술	리액트 내에서 SPA 형태로 프론트엔드 프로그램을 구현할 수 있다.	매우우수(5), 우수(4), 보통(3), 미흡(2), 매우미흡(1)	10	
	QueryString 파라미터를 처리할 수 있다.	매우우수(5), 우수(4), 보통(3), 미흡(2), 매우미흡(1)	10	
	Path 파라미터를 처리할 수 있다.	매우우수(5), 우수(4), 보통(3), 미흡(2), 매우미흡(1)	10	
	백엔드와의 연동을 위한 비동기 처리를 구현할 수 있다.	매우우수(5), 우수(4), 보통(3), 미흡(2), 매우미흡(1)	10	
	비동기 처리의 결과값을 리덕스를 활용하여 관리할 수 있다.	매우우수(5), 우수(4), 보통(3), 미흡(2), 매우미흡(1)	10	
	컴포넌트 최적화를 위한 처리를 수행할 수 있다.	매우우수(5), 우수(4), 보통(3), 미흡(2), 매우미흡(1)	10	
	Hook을 사용하여 상태값이 변경되었을 경우 필요한 데이터를 백엔드에 요청할 수 있다.	매우우수(5), 우수(4), 보통(3), 미흡(2), 매우미흡(1)	10	
	백엔드의 결과값을 필요한 형식에 맞춰 재구성할 수 있다.	매우우수(5), 우수(4), 보통(3), 미흡(2), 매우미흡(1)	10	
	백엔드로부터의 응답 결과를 Chart 형식으로 출력할 수 있다.	매우우수(5), 우수(4), 보통(3), 미흡(2), 매우미흡(1)	10	
태도	성실하고 철저한 자세로 작업물을 완성할 수 있다.	매우우수(5), 우수(4), 보통(3), 미흡(2), 매우미흡(1)	10	
총점			100	

채점기준

리액트 프로그래밍~covid19 보고서

박세영

목차

- 구현 결과
- 소스 코드
- 소감

구현 결과 ~초기 화면

Covid19 현황

2020-02-17



~

2022-05-31



검색

일일 확진자 | 누적 확진자 | 격리 환자 | 격리 해제 | 누적 격리 해제 | 사망자 | 누적 사망자

구현 결과 ~날짜 선택 후 검색버튼 클릭 시

localhost:3000/confirmed?gteDate=2022-02-17<eDate=2022-05-24

Covid19 현황

2022-02-17



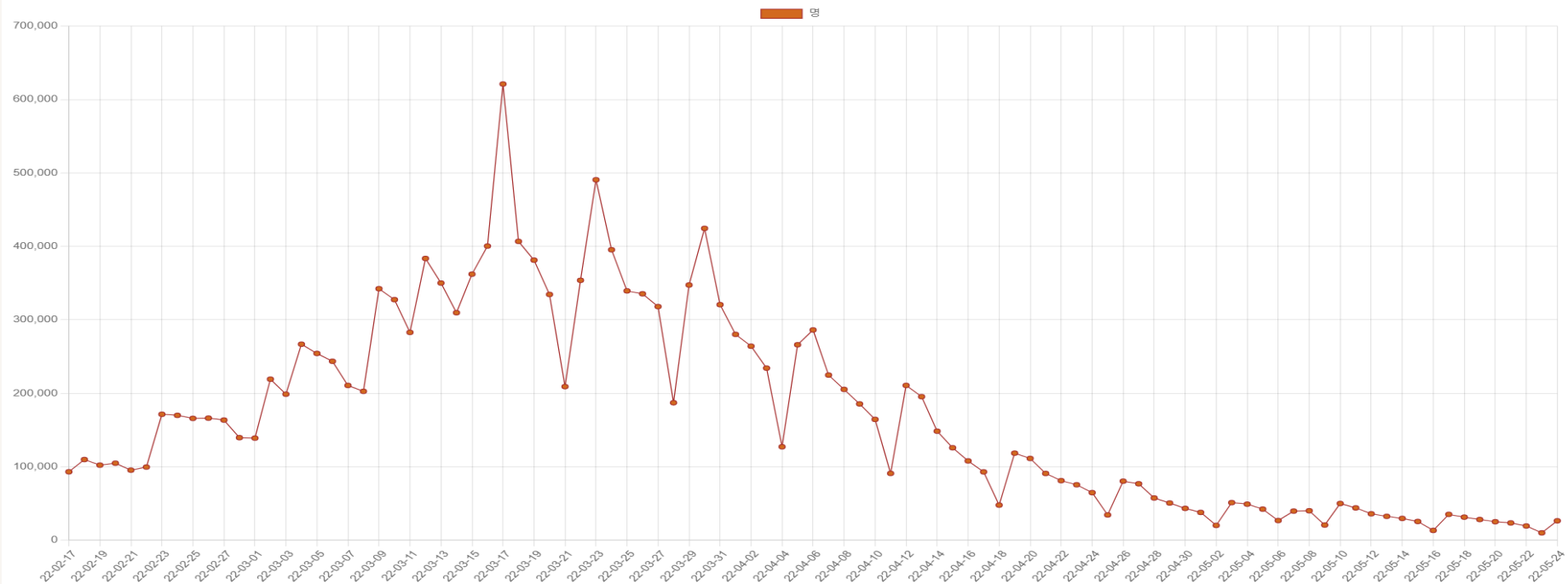
~

2022-05-24



검색

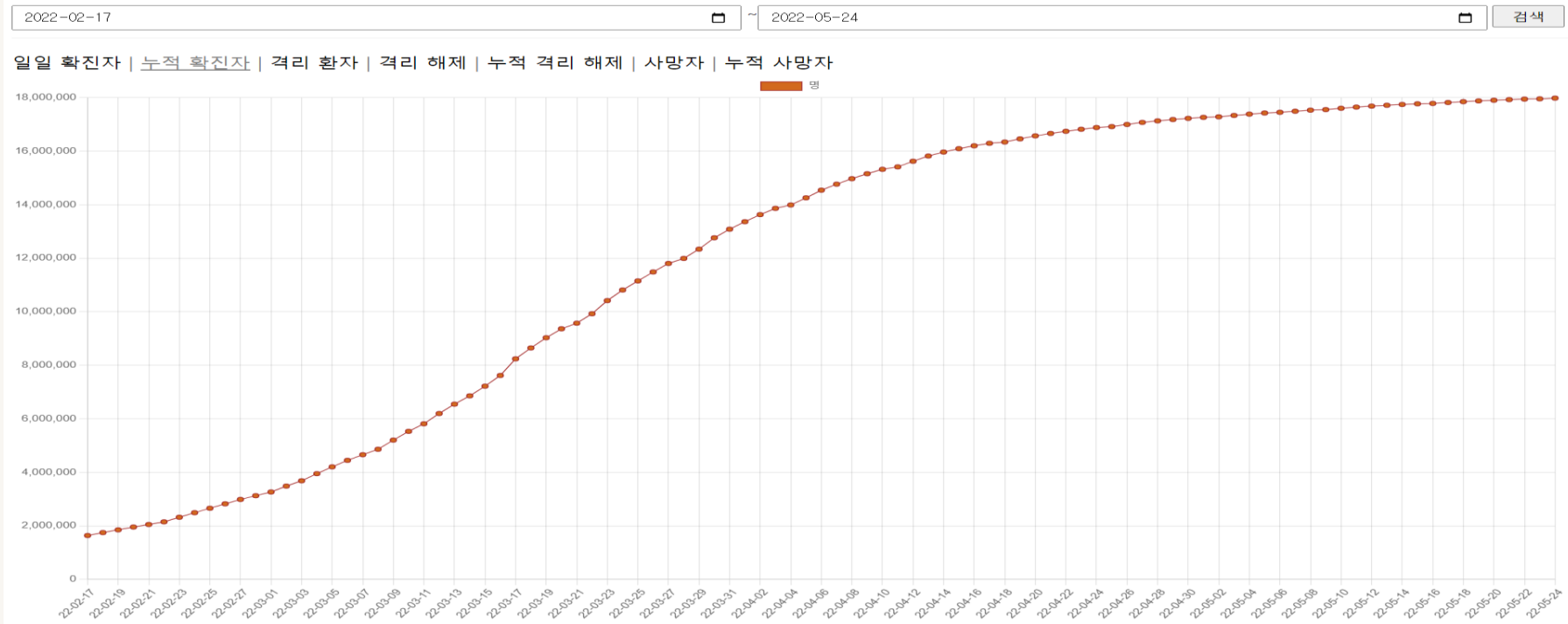
일일 확진자 | 누적 확진자 | 격리 환자 | 격리 해제 | 누적 격리 해제 | 사망자 | 누적 사망자



구현 결과 ~ 누적 확진자

localhost:3000/confirmed_acc?gteDate=2022-02-17<eDate=2022-05-24

Covid19 현황



구현 결과 ~ 격리 환자

localhost:3000/active?gteDate=2022-02-17<eDate=2022-05-24

Covid19 현황

2022-02-17



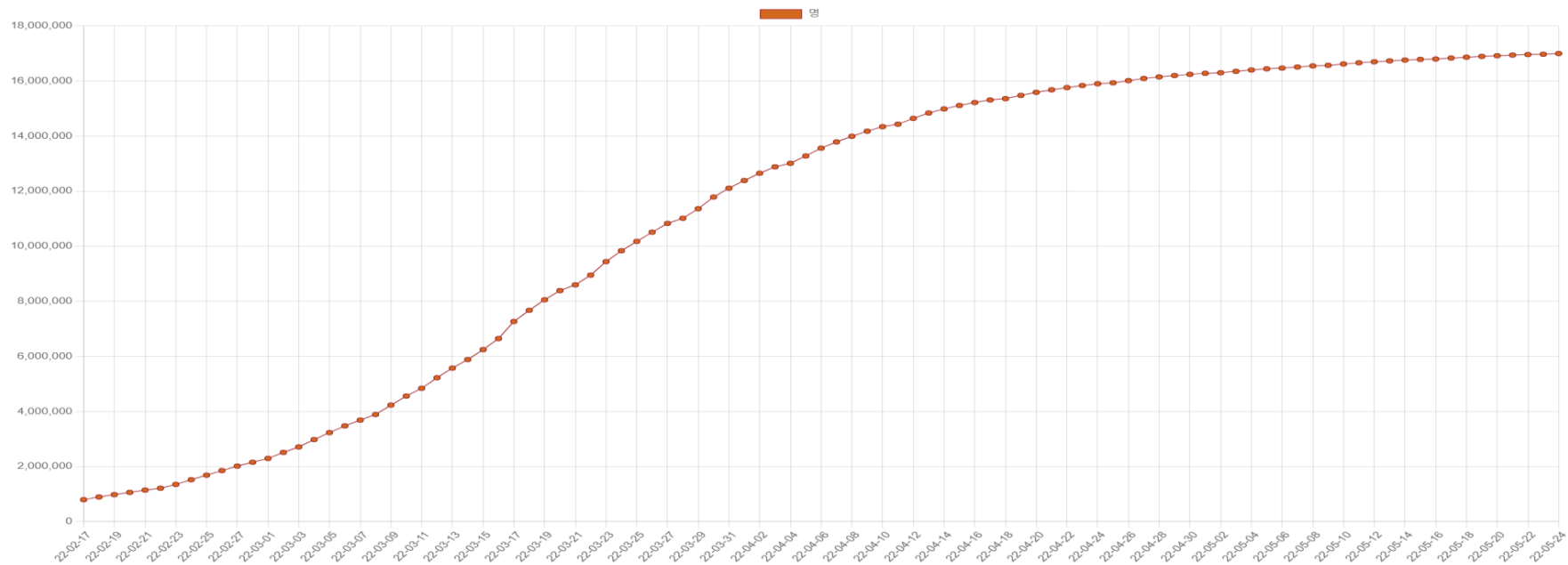
~

2022-05-24



검색

일일 확진자 | 누적 확진자 | 격리 환자 | 격리 해제 | 누적 격리 해제 | 사망자 | 누적 사망자



구현 결과~ 격리 해제

localhost:3000/released?gteDate=2022-02-17<eDate=2022-05-24

Covid19 현황

2022-02-17

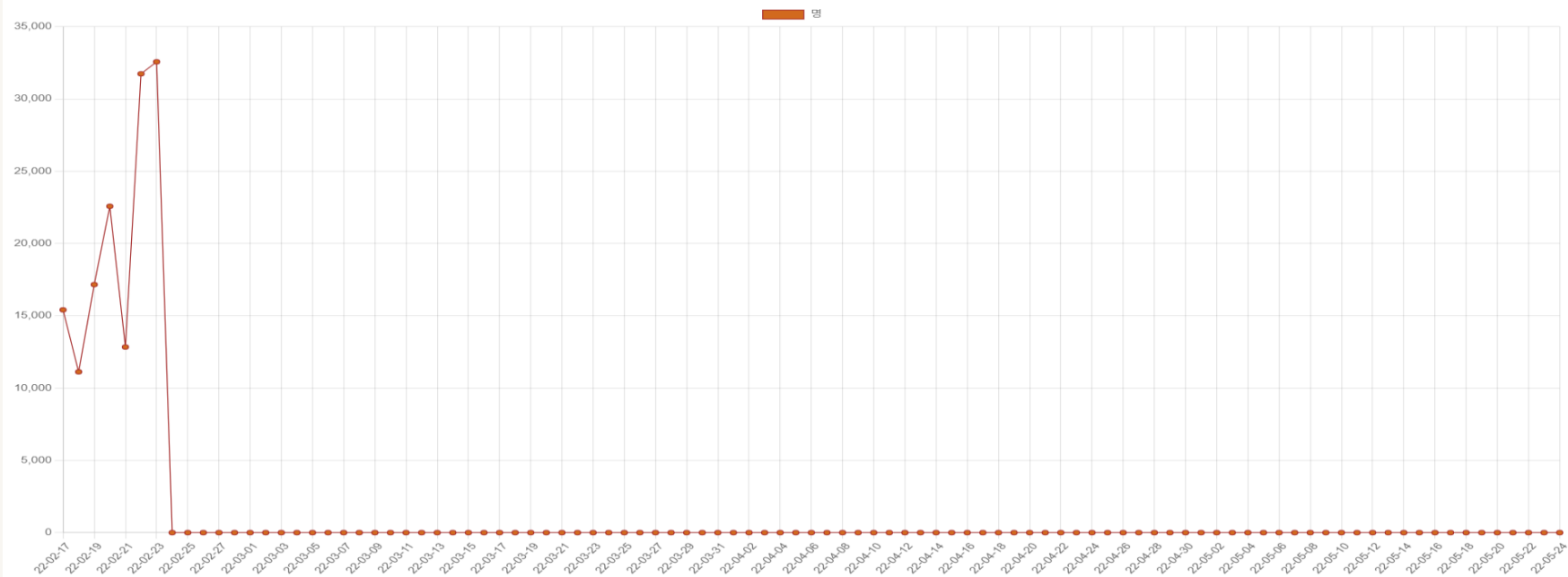


~ 2022-05-24



검색

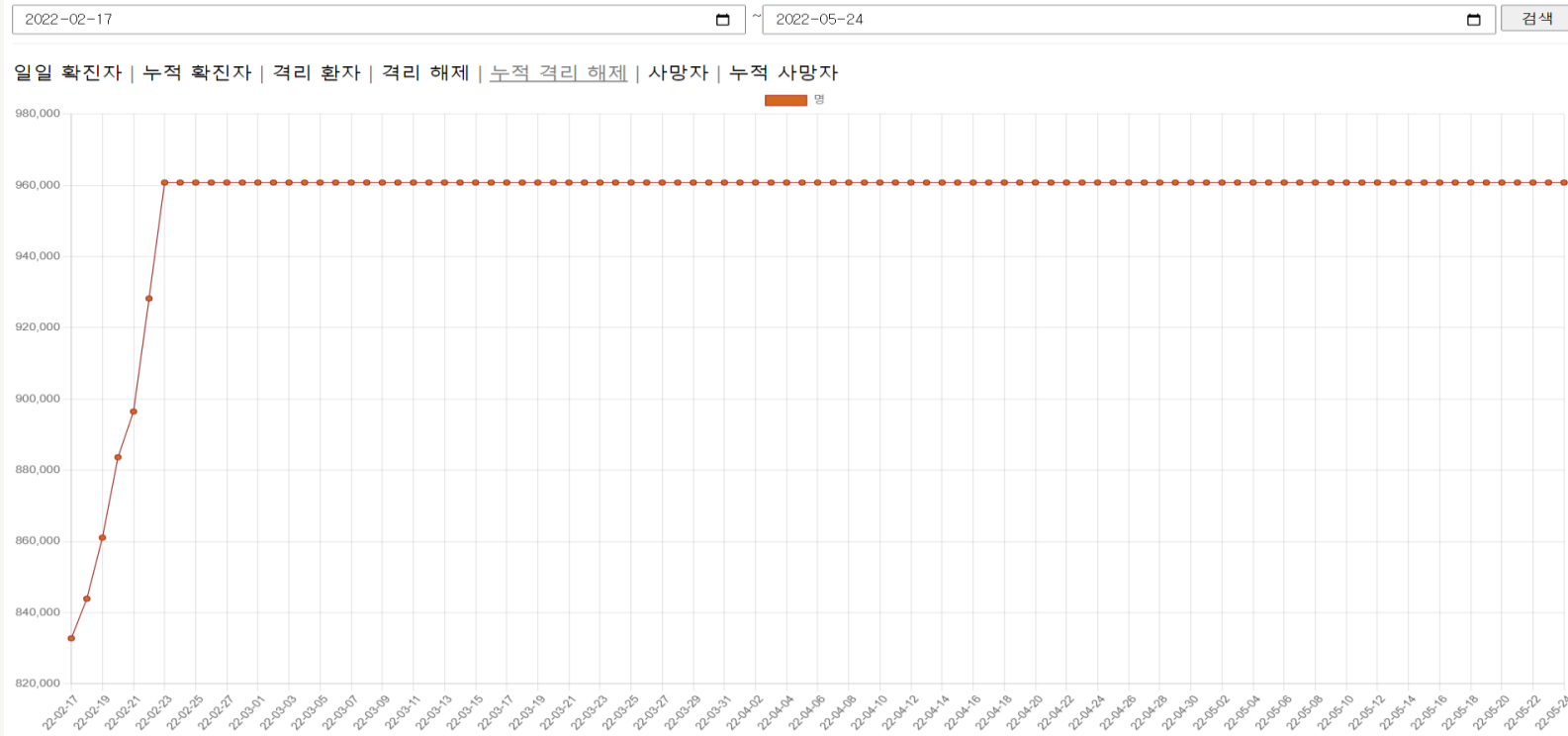
일일 확진자 | 누적 확진자 | 격리 환자 | 격리 해제 | 누적 격리 해제 | 사망자 | 누적 사망자



구현 결과~ 누적 격리 해제

localhost:3000/released_acc?gteDate=2022-02-17<eDate=2022-05-24

Covid19 현황



구현 결과~ 사망자

localhost:3000/death?gteDate=2022-02-17<eDate=2022-05-24

Covid19 현황

2022-02-17



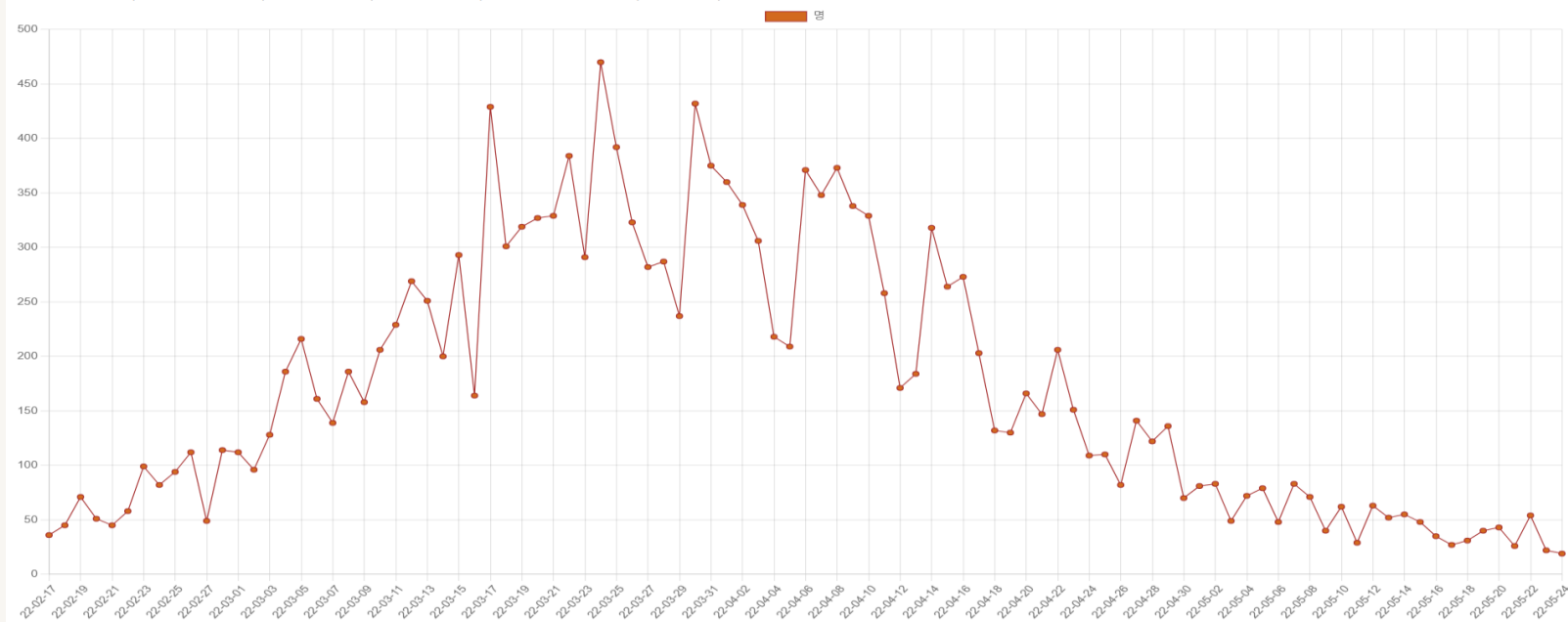
~

2022-05-24



검색

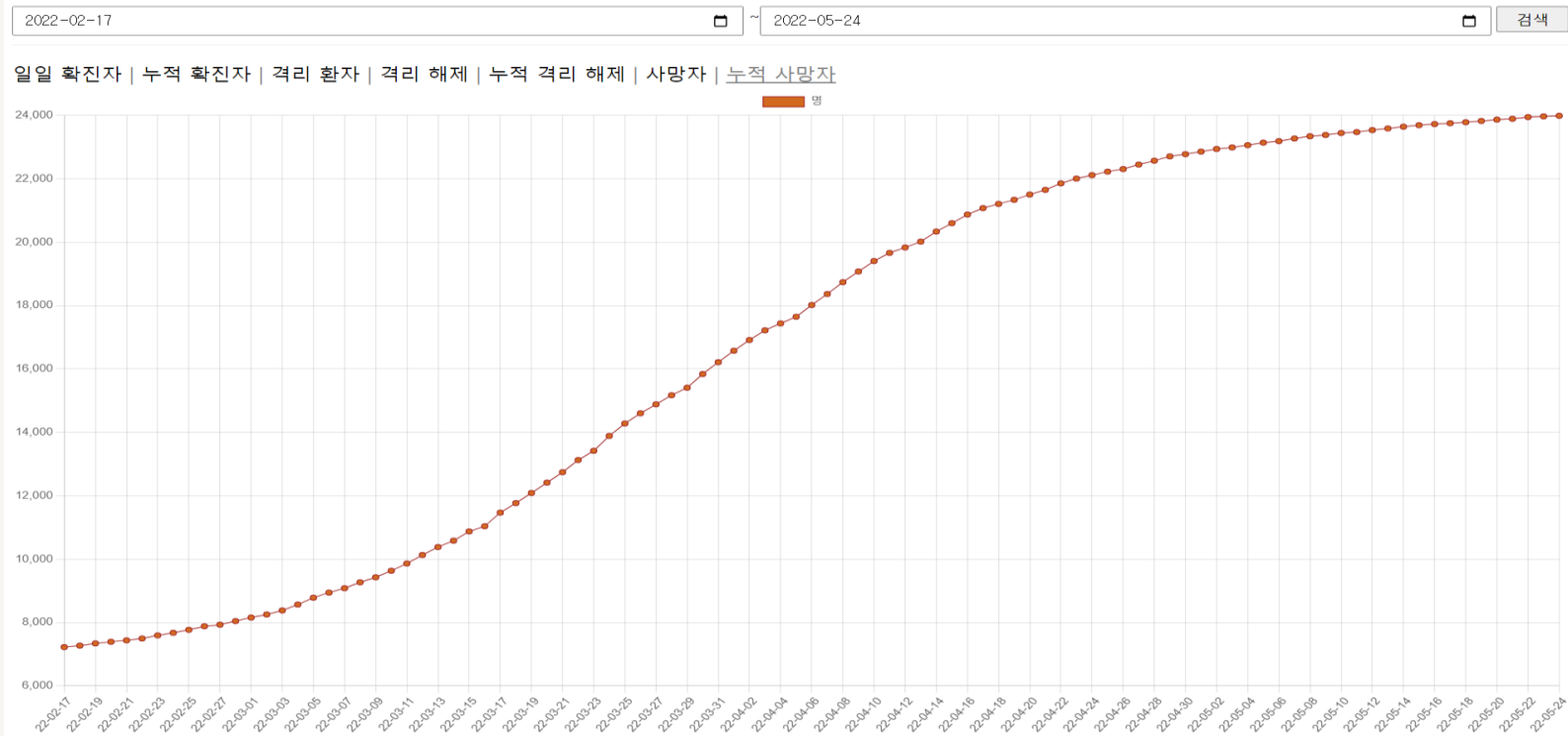
일일 확진자 | 누적 확진자 | 격리 환자 | 격리 해제 | 누적 격리 해제 | [사망자](#) | 누적 사망자



구현 결과~ 누적 사망자

localhost:3000/death_acc?gteDate=2022-02-17<eDate=2022-05-24

Covid19 현황



소스 코드

index.js

```
/**
 * @filename: index.js
 * @description: React 초기화 파일
 * @author: 박세영(qkrtpdud9899@gmail.com)
 */

import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import {BrowserRouter} from 'react-router-dom';

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

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

Meta.js

```
import React, { memo } from 'react';
import { Helmet, HelmetProvider } from 'react-helmet-async';

import sample from './assets/img/sample.png';

const Meta = memo((props) => {
  return (
    <HelmetProvider>
      <Helmet>
        <meta charset="utf-8"/>
        <title>{props.title}</title>
        { /*SEO 태그 */ }
        <meta name="description" content={props.description}/>
        <meta name="keywords" content={props.keywords}/>
        <meta name='author' content={props.author}/>
      </Helmet>
    </HelmetProvider>
  );
});
```

```

    <meta property='og:type' content='website' />
    <meta property='og:title' content={props.title}/>
    <meta property='og:description' content={props.description}/>
    <meta property='og:image' content={props.image}/>
    <meta property='og:url' content={props.url}/>
    <link rel='shortcut icon' href={props.image} type='image/png' />
    <link rel='icon' href={props.image} type='image/png' />
  </Helmet>
</HelmetProvider>
);
});

Meta.defaultProps = {
  title: 'React 코로나19 시험',
  description: 'React.js로 만든 Redux활용 코로나19 시험입니다.',
  keywords: 'React,Redux,covid19',
  author: 'Young',
  image: sample,
  url: window.location.href
};

export default Meta

```

App.js

```

/**
 * @filename: App.js
 * @description: 페이지 시작점
 * @author: 박세영(qkrtpdud9899@gmail.com)
 */

import React from 'react';
import {Routes, Route} from 'react-router-dom';

import Covid19 from './pages/Covid19';
import Top from './components/Top';

function App() {
  return (
    <div>
      <Top/>
      <Routes>
        <Route path='/:api' element={<Covid19/>}/>
      </Routes>
    </div>
  );
}

export default App;

```

/slices/Covid19Slice.js

```
/**
 * @filename: Covid19Slice.js
 * @description: Redux 구현체
 * @author: 박세영(qkrtpdud9899@gmail.com)
 */

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

//비동기 처리 함수 구현- ajax요청을 위한 함수
export const getList = createAsyncThunk(
  "covid19/getList",
  async (payload, { rejectWithValue }) => {
    let result = null;
    try {
      result = await axios.get("http://localhost:3001/covid19", {
        params: {
          date_gte: payload.date_gte,
          date_lte: payload.date_lte
        }
      });
    } catch (e) {
      result = rejectWithValue(e.response);
    }
    return result; //ajax 연동 결과를 리턴
  }
);

//ajax의 처리결과를 관리할 상태값 정의
const covid19Slice = createSlice({
  name: "covid19",
  initialState: {
    data: null,
    loading: false,
    error: null,
  },
  reducers: {
  },
  extraReducers: {
    [getList.pending]: (state, { payload }) => {
      return { ...state, loading: true };
    },
    [getList.fulfilled]: (state, { payload }) => {
      return {
        data: payload?.data,
        loading: false,
        error: null,
      };
    }
  }
});
```



```

    };
  },

  [getList.rejected]: (state, { payload }) => {
    return {
      data: payload?.data,
      loading: false,
      error: {
        code: payload?.status ? payload.status : 500,
        message: payload?.statusText ? payload.statusText : "Server Error",
      },
    };
  },
},
});

//리듀서 객체 내보내기
export default covid19Slice.reducer;

```

Store.js

```

/**
 * @filename: Store.js
 * @description: Redux Store
 * @author: 박세영(qkrtpdud9899@gmail.com)
 */

import { configureStore, getDefaultMiddleware } from '@reduxjs/toolkit';

import Covid19Slice from './slices/Covid19Slice';

const store= configureStore({
  reducer: {
    covid19: Covid19Slice
  },
  middleware: [...getDefaultMiddleware({serializableCheck: false})],

  devtools: true
});

export default store;

```

/components/Top.js

```

import React, { memo, useCallback } from "react";
import { useNavigate } from "react-router-dom";

```

```
import styled from "styled-components";
import { getList } from "../slices/Covid19Slice";
import dayjs from "dayjs";
import MenuLink from "../components/MenuLink";
import { useDispatch } from "react-redux";

const Form = styled.form`
  position: sticky;
  display: flex;
  top: 0;
  background-color: #fff;
  border-top: 1px solid #eee;
  border-bottom: 1px solid #eee;
  padding: 10px 0;
  margin: 0;
  margin-bottom: 20px;
  input,
  button {
    display: block;
    margin-right: 5px;
    font-size: 16px;
    padding: 0 10px;
    height: 30px;
  }
  input {
    flex: 1;
  }
  button {
    width: 70px;
    flex: none;
  }
`;

const Top = memo(() => {
  const navigate = useNavigate();
  const dispatch = useDispatch();

  let [gteDate, setGteDate] = React.useState('2020-02-17');
  let [lteDate, setLteDate] = React.useState('2022-05-31');

  const onDateGteChange = (
    (e) => {
      setGteDate(e.target.value);
    }
  );

  const onDateLteChange = (
    (e) => {
      setLteDate(e.target.value);
    }
  );

  const onSearchSubmit = useCallback(
    (e) => {
      e.preventDefault();
      if(gteDate > lteDate){
```

```

    alert('시작 날짜가 더 빨라야합니다');
  }else if(gteDate < '2020-02-17' || lteDate > '2022-05-31'){
    alert('2020-02-17부터 2022-05-31까지의 날짜만 선택 가능합니다.');
```

}

```

    navigate(`/confirmed?gteDate=${gteDate}&lteDate=${lteDate}`);
  },
  [navigate, gteDate, lteDate]
);
const lteOneDay= dayjs(lteDate).add(1, 'd').format('YYYY-MM-DD');

React.useEffect(()=>{
  dispatch(getList({date_gte: gteDate, date_lte: lteOneDay}))
},[dispatch, gteDate, lteOneDay]);

return (
  <div>
    <h1> Covid19 현황 </h1>
    <br />
    <Form onSubmit={onSearchSubmit} name="dateForm">
      <input
        type="date"
        name="gte"
        value={gteDate}
        onChange={onDateGteChange}
      />
      ~
      <input
        type="date"
        name="lte"
        value={lteDate}
        onChange={onDateLteChange}
      />
      <button type="submit" >
        검색
      </button>
    </Form>

    <nav>
      <MenuLink
        to={`/confirmed?gteDate=${gteDate}&lteDate=${lteDate}`}
      >
        일일 확진자
      </MenuLink>
      <MenuLink
        to={`/confirmed_acc?gteDate=${gteDate}&lteDate=${lteDate}`}
      >
        누적 확진자
      </MenuLink>
      <MenuLink to={`/active?gteDate=${gteDate}&lteDate=${lteDate}`} >
        격리 환자
      </MenuLink>
      <MenuLink
        to={`/released?gteDate=${gteDate}&lteDate=${lteDate}`}
      >

```

```

        격리 해제
      </MenuLink>
      <MenuLink
        to={` /released_acc?gteDate=${gteDate}&lteDate=${lteDate}`}
      >
        누적 격리 해제
      </MenuLink>
      <MenuLink to={` /death?gteDate=${gteDate}&lteDate=${lteDate}`} >
        사망자
      </MenuLink>
      <MenuLink
        to={` /death_acc?gteDate=${gteDate}&lteDate=${lteDate}`}
      >
        누적 사망자
      </MenuLink>
    </nav>
  </div>
);
});

export default Top;

```

/components/LineChartView.js

```

import React from "react";
import "chart.js/auto";

import {
  Chart as ChartJS,
  CategoryScale,
  LinearScale,
  BarElement,
  Title,
  Tooltip,
  Legend,
} from "chart.js";

import { Line } from "react-chartjs-2";

ChartJS.register(
  CategoryScale,
  LinearScale,
  BarElement,
  Title,
  Tooltip,
  Legend
);

const LineChartView = ({ tgData }) => {

```

```

const options = {
  responsive: true,
  plugins: {
    legend: {
      position: "top",
    },
  },
};

const data = {
  labels: tgData.date,
  datasets: [
    {
      label: "명",
      backgroundColor: "chocolate",
      borderColor: "brown",
      borderWidth: 1,
      data: tgData.num,
    },
  ],
};

return <Line options={options} data={data} />;
};

LineChartView.defaultProps = {
  tgData: {
    date: [],
    num: [],
  },
};

export default React.memo(LineChartView);

```

/pages/Covid19.js

```

import React, { memo } from "react";
import { useParams } from "react-router-dom";
import { useSelector } from "react-redux";

import Spinner from "../components/Spinner";
import ErrorView from "../components/ErrorView";
import LineChartView from "../components/LineChartView";

const Covid19 = memo(() => {
  //필드명
  const { api } = useParams();
  //lte=최대값, gte=최소값
  const { data, loading, error } = useSelector((state) => state.covid19);

```

```
//선택한 날짜의 데이터
const [tgData, setTgData]= React.useState();

React.useEffect(()=>{
  const getDate= {
    date: [],
    num: [],
    name: api
  }
  data && data.forEach((v, i) => {
    getDate.date.push(v.date.slice(2,10));
    getDate.num.push(v[api]);
  });
  setTgData(getDate);
  console.log(getDate);
},[data, api])

return (
  <div>
    <Spinner visible={loading} />
    {error ? (
      <ErrorView error={error} />
    ) : (
      tgData && <LineChartView tgData={tgData}/>
    )}
  </div>
);
});

export default Covid19;
```

components폴더 안의 **ErrorView.js**와 **MenuLink.js**, 그리고 **Spinner.js** hooks 폴더안의 **useQueryString.js**는 이전 실습과 같은 코드입니다

소감

- 처음 문제를 3일정도 붙잡고 풀다 저의 부족한 실력을 탓하며 포기했습니다. 하지만 감사하게도 다른 수강생분들께서 할수있다며 응원해주시고 제가 이해가 가지 않는 부분을 차근차근 설명해주셔서 결국 완성을 했습니다.

Covid19 현황

연도-월-일 ~ 연도-월-일

2020-02-17T00:00:00Z
20
2020-02-18T00:00:00Z
20
2020-02-19T00:00:00Z
50
2020-02-20T00:00:00Z
65
2020-02-21T00:00:00Z

- 제가 처음 혼자 풀다가 멈췄던 부분 입니다.
이 이상 어떻게 해야할지 감이 오지않아 무작정 코드를 작성했고 결국 3일동안 더이상 진도가 나가지 않아 포기했었습니다.

```
return (  
  <div>  
    <Spinner visible={loading} />  
    {error ? (  
      <ErrorView error={error} />  
    ) : (  
      data && (  
        data.map((v, i) => (  
          <div>  
            <div key={i}>  
              <p>{v.date}</p>  
              <p>{v.active}</p>  
            </div>  
          </div>  
        )  
      )  
    )  
  )  
);
```

- 다른 수강생분의 설명을 차근차근 들으며 코드의 흐름을 감을 잡았습니다.
두루뭉술하게만 뚱뚱 떠있던 여러 생각들이 정리되었고 처음 백엔드에 보내야할 url값을 확인하고 그 주소값을 어떻게 활용해서 써야하는지 dispatch와 useSelector를 이용해 백엔드에 보내야할 url을 만들고 그 url과 눈에 보이는 주소창은 (localhost:3000~~) 같을 필요가 없다고 설명을 들었습니다.
처음에는 아 그런가보다~ 하고 넘겼고 두번째, 세번째 계속 보다보니 이해가 100% 다 되진 않았지만 저 설명을 듣고 제가 어떻게 코드를 배체하고 짜야하는지 서서히 눈에 보이기 시작했습니다.

```
React.useEffect(()=>{
  const getDate= {
    date: [],
    num: [],
    name: api
  }
  data && data.forEach((v, i) => {
    getDate.date.push(v.date.slice(2,10));
    getDate.num.push(v[api]);
  });
  setTgData(getDate);
  console.log(getDate);
},[data, api])
```

- 값이 출력이 되지않아 원인을 찾던중 이 코드의 num값이 undefined로 출력되는것을 확인했습니다. 확인후 원인을 찾기 위해 여러곳을 되짚어 보았습니다.

```
<MenuLink
  to={` /covid19?query=confirmed_gte=${gte}&confirmed_lte=${lte}`}
>
  ...
</MenuLink>
```

그러다 발견한 MenuLink 구문에서 잘못 전달되고 있다는것을 확인했고 알맞게 바꾸어 주었습니다.

```
<MenuLink
  to={` /confirmed?gteDate=${gteDate}&lteDate=${lteDate}`}
>
  ...
</MenuLink>
```


- 그 외에 작은 오타와 차트데이터를 넣어주는 부분에서 헤맸지만 잘 해결되었습니다.
- 다른 수강생분들이 설명을 간단히 해주시고 이후 15시간동안 다른 수강생분들의 설명을 참고해 결과물을 만들었습니다. 다른 수강생분들의 코드를 참고했기때문에 많이 코드가 많이 비슷합니다. 하지만 저 혼자 망망대해에 있는것처럼 느껴져 포기를 했지만 다른분들의 도움으로 주어진 시간안에 끝까지 해보았다는게 개인적으로 많이 뿌듯합니다. 한번도 포기했던걸 다시 붙잡고 도전한적이 없었던 저에게 큰 의미로 남은 시간이였습니다.
- 다른 수강생분들의 코드를 참고하면서 혼자서는 방향을 잡을 수 없었던 부분을 방향을 잡아가며 세세히 물어보지 못한부분을 혼자 고민하며 이해를 해가며 작성했습니다. 아직 많이 부족하지만 난해하던 리액트와 리덕스를 좀 더 이해할수 있는 시간이였고 복습을 어떤 부분을 해야할지 확인할수 있었습니다.