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}/>
            <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 내보네기
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.js와 MenuLink.js, 그리고 Spinner.js hooks 폴더안의 useQueryString.js는 이전 실습과 같은 코드입니다