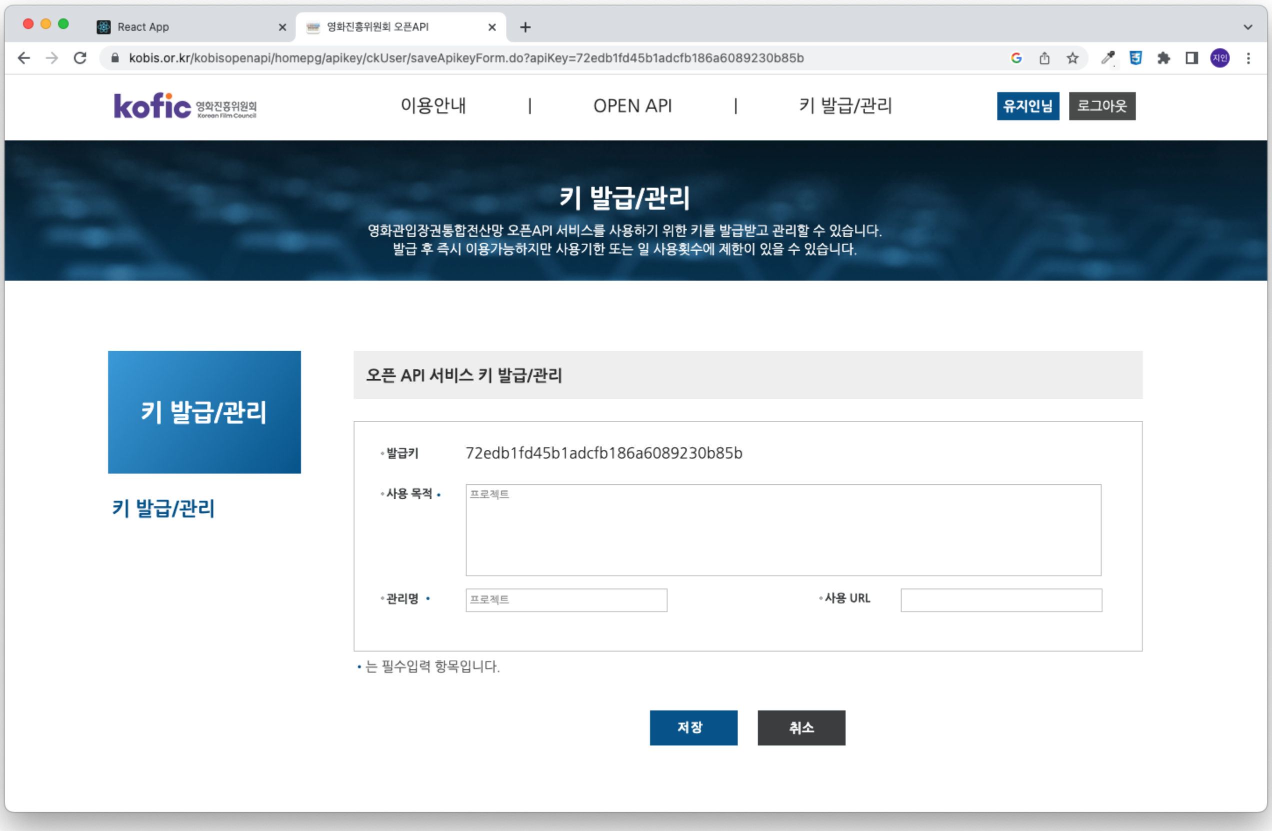
유지인 통합구현 평가문제

영화진흥위원회 OpenAPI 인증키 발급화면


React App
영화진흥위원회 오픈API
kobis.or.kr/kobisopenapi/homepg/apikey/ckUser/saveApikeyForm.do?apiKey=72edb1fd45b1adcfb186a6089230b85b

kofic 영화진흥위원회 Korean Film Council
이용안내 | OPEN API | 키 발급/관리
유지인님
로그아웃

키 발급/관리

영화관입장권통합전산망 오픈API 서비스를 사용하기 위한 키를 발급받고 관리할 수 있습니다.
발급 후 즉시 이용가능하지만 사용기한 또는 일 사용횟수에 제한이 있을 수 있습니다.

오픈 API 서비스 키 발급/관리

발급키	72edb1fd45b1adcfb186a6089230b85b		
사용 목적 •	프로젝트		
관리명 •	프로젝트	사용 URL	

• 는 필수입력 항목입니다.

App.js

```
import React, { memo, useState, useEffect, useCallback, useMemo } from
"react";
import styled from "styled-components";

import { useSelector, useDispatch } from 'react-redux'
import { getMovieRank } from "./slices/MovieRankSlice";

// 로딩바 컴포넌트
import Spinner from './components/Spinner';
// 에러정보를 표시하기 위한 컴포넌트
import ErrorView from './components/ErrorView';
// 그래프 표시르 위한 컴포넌트
import BarChartView from './components/BarChartView';
// 미디어쿼리
import mq from './MediaQuery';
// 날짜 처리 라이브러리
import dayjs from "dayjs";

const Container = styled.div`
```

```
    ${mq.minWidth('md')`
        display: flex;
        flex-direction : row;
        flex-wrap: nowrap;

        .flex-item {
            width: 50%;
            box-sizing: border-box;
            padding: 10px;
        }
    `}
`;

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

  const dispatch = useDispatch();

  const [targetDt, setTargetDt] = useState(dayjs().add(-1,
'd').format('YYYY-MM-DD'));

  useEffect(() => {
    dispatch(getMovieRank({
      targetDt: targetDt.replaceAll('-', '')
    }));
  },[targetDt]);

  // 드롭다운 이벤트
  const onDataChange = useCallback((e) => {
    e.preventDefault();
    setTargetDt(e.target.value);
  },[]);

  // data가 변경되었을 때, 사이드 이펙트를 처리하여 그래프에 적용할 데이터를 생성한다.
  const { movieNm, audiCnt } = useMemo(() => {
    const newData = { movieNm: [], audiCnt: []};

    if (data) {
      data.boxOfficeResult.dailyBoxOfficeList.forEach((v, i) => {
        newData.movieNm.push(v.movieNm);
        newData.audiCnt.push(v.audiCnt);
      });

      console.log(newData);
    }

    return newData;
  },[data]);

  return (
    <div>
      <Spinner loading={loading} />
      <h1>영화진흥위원회 박스오피스 순위</h1>

      <form>
        <input type="date" className="form-control" placeholder="연도-월-일"
onChange={onDataChange}/>
      </form>

      {error? <ErrorView error={error} /> : (
            <Container>

                <div className="flex-item">
                    <table border='1'>
                        <thead>
                            <tr>
                                <th>순위</th>
                                <th>영화제목</th>
                                <th>관객수</th>
                                <th>개봉일</th>
                            </tr>
                        </thead>
                        <tbody>
                            {data &&
data.boxOfficeResult.dailyBoxOfficeList.map((v, i) => {
                                return (
                                    <tr key={i}>
                                        <td>{v.rank}</td>
                                        <td>{v.movieNm}</td>
                                        <td>
{Number(v.audiCnt).toLocaleString()}</td>
                                        <td>{v.openDt}</td>
                                    </tr>
                                )
                            })}
                        </tbody>
                    </table>
                </div>
                <div className="flex-item">
                    <BarChartView labels={movieNm} dataset={audiCnt}
legend='관람객 수' />
                </div>
            </Container>
        )}
    </div>
  );
});

export default App;
```

store.js

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

const store = configureStore({
    reducer: {
        MovieRankSlice: MovieRankSlice,
    }
});

export default store;
```

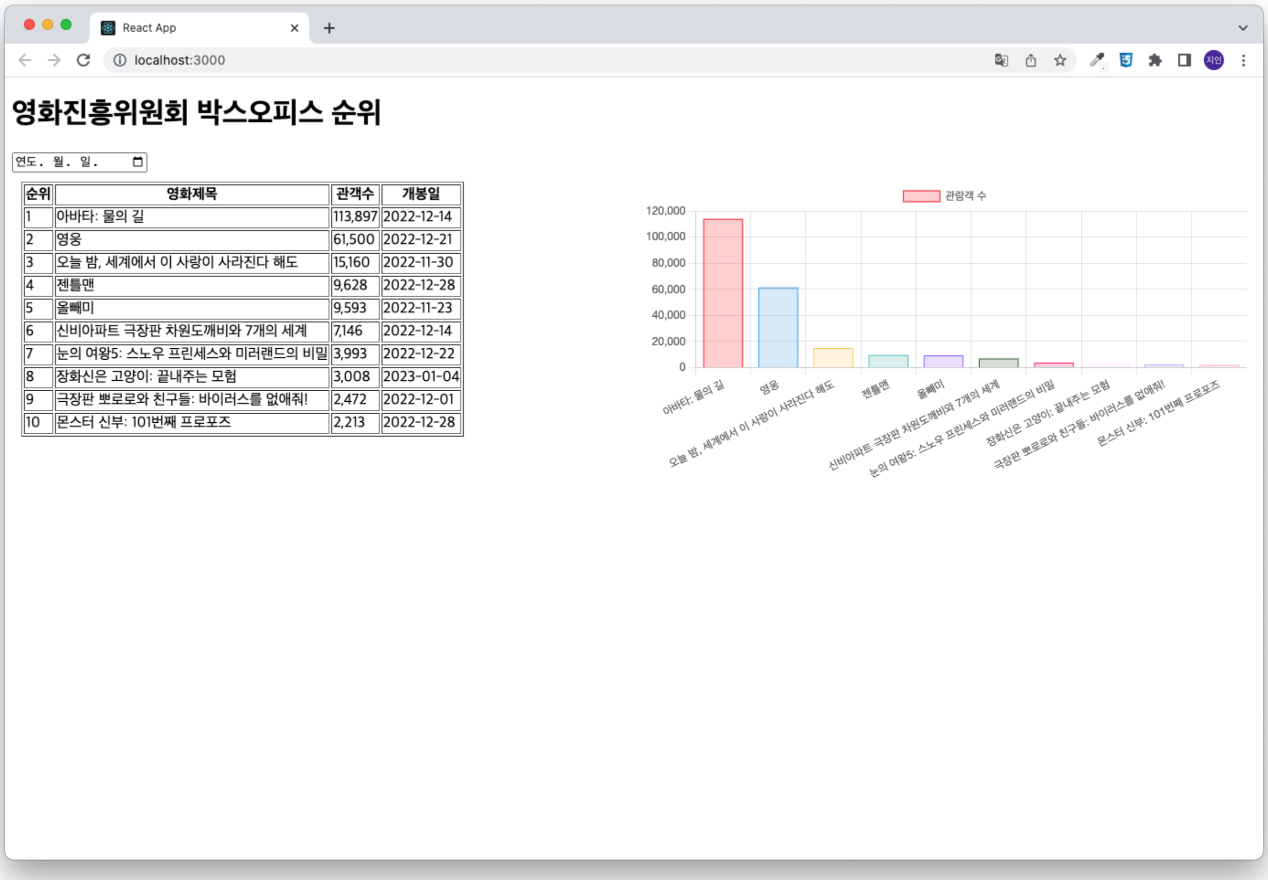
slices/MovieRankSlice.js

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

export const getMovieRank =
createAsyncThunk('MovieRankSlice/getMovieRank', async (payload, {
rejectWithValue }) => {
    let result = null;

    try {
        const response = await
axios.get(process.env.REACT_APP_KOBIS_API_URL, {
            params: {
                key: process.env.REACT_APP_KOBIS_API_KEY,
                targetDt: payload.targetDt
            }
        });
        result = response.data;

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

    return result;
});

const MovieRankSlice = createSlice({
    name: 'MovieRankSlice',
    initialState: {
        data: null,
        loading: false,
        error: null
    },
    reducers: {},
    extraReducers: {
        [getMovieRank.pending]: (state, { payload }) => {
            return {...state, loading: true }
        },
        [getMovieRank.fulfilled]: (state, { payload }) => {
            return {
                data: payload,
                loading: false,
                error: null
            }
        },
        [getMovieRank.rejected]: (state, { payload }) => {
            return {
                ...state,
                loading: false,
                error: {
                    code: payload.status ? payload.status : 500,
                    message: payload.statusText ? payload.statusText :
'Server Error'
                }
            }
        }
    },
});

export default MovieRankSlice.reducer;
```

components/BarChartView.js

```
import React, { memo } from 'react';

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

import { Bar } from 'react-chartjs-2';

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

const BarChartView = memo(({ labels, dataset, legend}) => {
    /** 그래프 옵션 */
    const options = {
        indexAxis: 'x',
        responsive: true
    };

    /** chart에 표시될 데이터 (막대그래프용) */
    const data = {
        labels: labels,
        datasets: [{
            label: legend,
            backgroundColor:['rgba(255, 0, 0,0.2)', 'rgba(54, 162, 235,
0.2)', 'rgba(255, 206, 86, 0.2)', 'rgba(75, 192, 192, 0.2)', 'rgba(153,
102, 255, 0.2)', 'rgba(53, 100, 50, 0.2)', 'rgba(250, 10, 100, 0.2)',
'rgba(254, 224, 255, 0.2)', 'rgba(183, 183, 234, 0.2)', 'rgba(255, 178,
195, 0.2)'],
            borderColor: ['rgba(255, 0, 0,1)', 'rgba(54, 162, 235, 1)',
'rgba(255, 206, 86, 1)', 'rgba(75, 192, 192, 1)', 'rgba(153, 102, 255,
1)', 'rgba(53, 100, 50, 1)', 'rgba(250, 10, 100, 1)', 'rgba(254, 224, 255,
1)', 'rgba(183, 183, 234, 1)', 'rgba(255, 178, 195, 1)'],
            borderWidth: 1,
            data: dataset,
        }]
    };

    return ((labels && dataset) && <Bar data={data} options={options} />);
});

BarChartView.defaultProps = {
    labels: [],
    dataset: [],
    legend: ''
}

export default BarChartView;
```

실행결과의 스크린샷

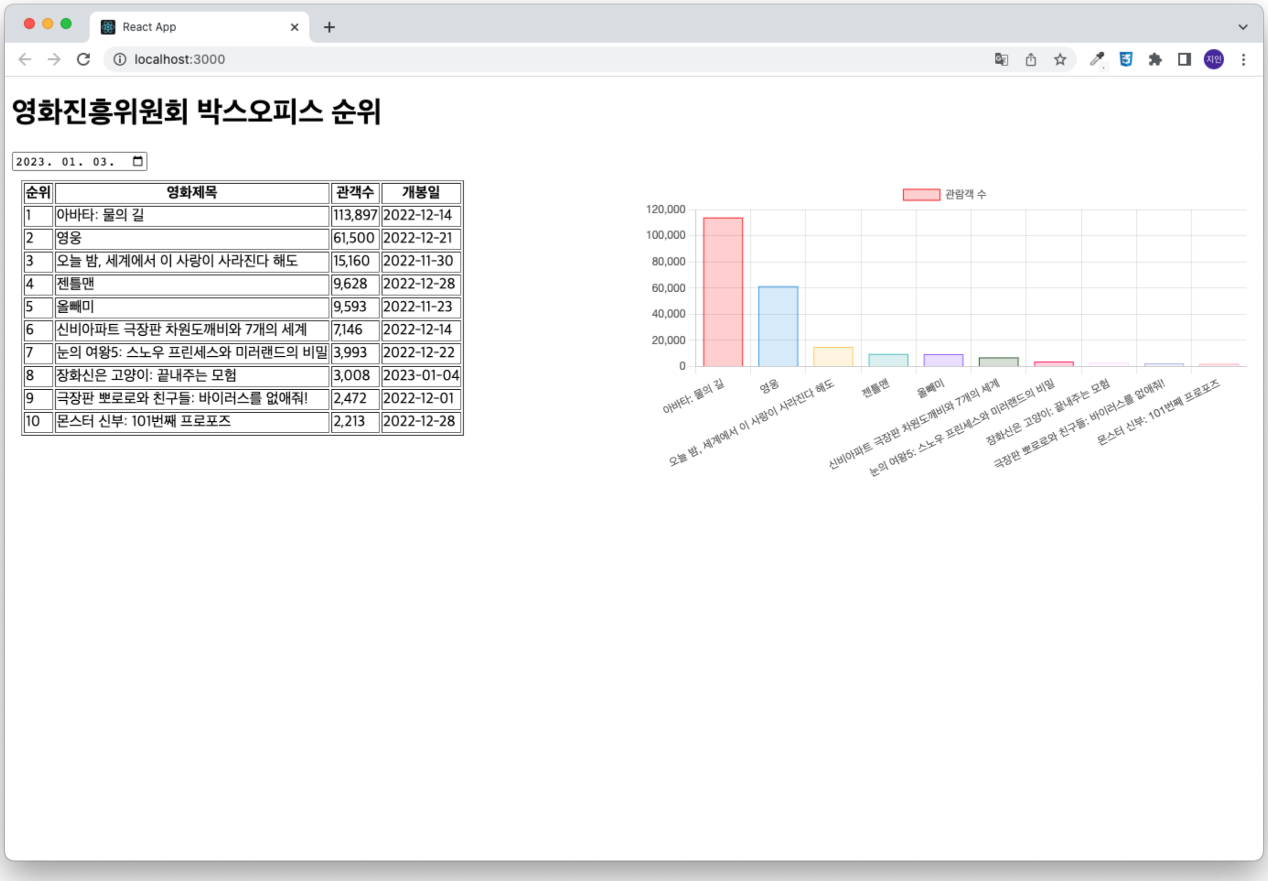

React App
localhost:3000

영화진흥위원회 박스오피스 순위

연도. 월. 일.

순위	영화제목	관객수	개봉일
1	아바타: 물의 길	113,897	2022-12-14
2	영웅	61,500	2022-12-21
3	오늘 밤, 세계에서 이 사랑이 사라진다 해도	15,160	2022-11-30
4	젠틀맨	9,628	2022-12-28
5	올빼미	9,593	2022-11-23
6	신비아파트 극장판 차원도깨비와 7개의 세계	7,146	2022-12-14
7	눈의 여왕5: 스노우 프린세스와 미러랜드의 비밀	3,993	2022-12-22
8	장화신은 고양이: 끝내주는 모험	3,008	2023-01-04
9	극장판 뽀로로와 친구들: 바이러스를 없애줘!	2,472	2022-12-01
10	몬스터 신부: 101번째 프로포즈	2,213	2022-12-28

관람객 수
120,000
100,000
80,000
60,000
40,000
20,000
0
아바타: 물의 길
영웅
오늘 밤, 세계에서 이 사랑이 사라진다 해도
젠틀맨
올빼미
신비아파트 극장판 차원도깨비와 7개의 세계
눈의 여왕5: 스노우 프린세스와 미러랜드의 비밀
장화신은 고양이: 끝내주는 모험
극장판 뽀로로와 친구들: 바이러스를 없애줘!
몬스터 신부: 101번째 프로포즈


React App
localhost:3000

영화진흥위원회 박스오피스 순위

2023. 01. 03.

순위	영화제목	관객수	개봉일
1	아바타: 물의 길	113,897	2022-12-14
2	영웅	61,500	2022-12-21
3	오늘 밤, 세계에서 이 사랑이 사라진다 해도	15,160	2022-11-30
4	젠틀맨	9,628	2022-12-28
5	올빼미	9,593	2022-11-23
6	신비아파트 극장판 차원도깨비와 7개의 세계	7,146	2022-12-14
7	눈의 여왕5: 스노우 프린세스와 미러랜드의 비밀	3,993	2022-12-22
8	장화신은 고양이: 끝내주는 모험	3,008	2023-01-04
9	극장판 뽀로로와 친구들: 바이러스를 없애줘!	2,472	2022-12-01
10	몬스터 신부: 101번째 프로포즈	2,213	2022-12-28

관람객 수
120,000
100,000
80,000
60,000
40,000
20,000
0
아바타: 물의 길
영웅
오늘 밤, 세계에서 이 사랑이 사라진다 해도
젠틀맨
올빼미
신비아파트 극장판 차원도깨비와 7개의 세계
눈의 여왕5: 스노우 프린세스와 미러랜드의 비밀
장화신은 고양이: 끝내주는 모험
극장판 뽀로로와 친구들: 바이러스를 없애줘!
몬스터 신부: 101번째 프로포즈