

API 연동 실습



Contents

- API 연동 실습 – 날씨 조회 앱
1. API
 2. API 연동 테스트
 3. (실습) 날씨 조회 앱 만들기

SECTION 1. API

◦ Axios

- API 연동을 위한 HTTP 통신 라이브러리

- axios 설치

```
npm install axios
```

- axios 사용법

- axios(config)

```
// POST 요청 전송
axios({
  method: 'post',
  url: '/user/12345',
  data: {
    firstName: 'Fred',
    lastName: 'Flint'
  }
});
```

- axios(url[, config]) – REST API

```
// GET 요청 전송 (기본값)
axios('/user/12345');

// POST 요청 전송
axios.post('/users', {
  firstName: 'Fred',
  lastName: 'Flint'
});
```

SECTION 1. API

- **OpenWeather API (<https://openweathermap.org/>)**

- 날씨 정보를 제공하는 API. 기능에 따라 유/무료 제공
- 회원 가입 후 api key를 발급받아 사용

- API > Current Weather Data > API doc

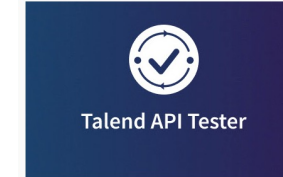
```
https://api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}
```



SECTION 2. API Test

- Talend API Tester – chrome 웹 확장 프로그램
 - 웹 스토어에서 api tester 검색하여 설치
 - REST API에 대응하는 다양한 HTTP METHOD 제공
 - GET: 데이터 조회
 - POST: 데이터 등록
 - PUT: 데이터 수정
 - DELETE: 데이터 삭제

확장 프로그램 ⓘ



Talend API Tester - Free Edition

추천

Visually interact with REST, SOAP and HTTP APIs.

★★★★★ 4,145 개발자 도구

SECTION 2. API Test

Current Weather API Test

- API URL 입력

METHOD: GET

SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]

https://api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}

length: 77 char(s) 89 byte(s)

QUERY PARAMETERS

<input checked="" type="checkbox"/>	q	=	{city name}	×	⋮
<input checked="" type="checkbox"/>	appid	=	{API key}	×	⋮

+ Add query parameter

- Query parameter 수정 후 Send

METHOD: GET

SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]

https://api.openweathermap.org/data/2.5/weather?q=seoul&appid=5bb4f17b9abe3688a33a27272c989cd7

length: 94 byte(s)

QUERY PARAMETERS

<input checked="" type="checkbox"/>	q	=	seoul	×	⋮
<input checked="" type="checkbox"/>	appid	=	5bb4f17b9abe3688a33a27272c989cd7	×	⋮

+ Add query parameter

5bb4f17b9abe3688a33a27272c989cd7

SECTION 2. API Test

Current Weather API Test

Response

Cache Detected - Elapsed Time: 611ms

200 OK

Send request (⌘ + Enter)

HEADERS ?

pretty ▼

Server: openresty
Date: Wed, 24 May 2023 13:22:14 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 524 bytes
Connection: keep-alive
X-Cache-Key: /data/2.5/weather?q=seoul
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST

▶ COMPLETE REQUEST HEADERS

BODY ?

pretty ▼

```
{
  coord : {
    lon : 126.9778,
    lat : 37.5683
  },
  weather : [
    { id : 804, main : "Clouds", description : "clouds" },
  ],
  base : "stations",
  main : {
    temp : 291.81,
    feels_like : 290.78,
    temp_min : 288.84,
    temp_max : 291.81,
    pressure : 1017,
    humidity : 40,
    sea_level : 1017,
    grnd_level : 1010
  },
}
```

weather.main – 기상 조건

main.temp – 기온

Unit Default: Kelvin (절대온도)
OK = -273.15도(섭씨)

SECTION 2. API Test

◦ Current Weather API Test

- optional parameter 제공

`units` optional Units of measurement. `standard`, `metric` and `imperial` units are available. If you do not use the `units` parameter, `standard` units will be applied by default. [Learn more](#)

`lang` optional You can use this parameter to get the output in your language. [Learn more](#)

- optional parameter 추가 후 Send

▼ QUERY PARAMETERS ↓

<input checked="" type="checkbox"/>	q	=	seoul	×	⋮
<input checked="" type="checkbox"/>	appid	=	5bb4f17b9abe3688a33a27272c989cd7	×	⋮
<input checked="" type="checkbox"/>	units	=	metric	×	⋮
<input checked="" type="checkbox"/>	lang	=	KR	×	⋮

SECTION 2. API Test

- Current Weather API Test
 - Response

```
weather : ▼ [  
  ▶ { id : 800, main : "Clear", description : "맑음",  
},  
  base : "stations",  
  main : ▼ {  
    temp : 16.84,  
    feels_like : 16.07,  
    temp_min : 14.69,  
    temp_max : 17.78,  
    pressure : 1016,  
    humidity : 57  
  },
```

weather.description – lang 지원

main 온도가 섭씨로 변경됨

SECTION 2. API Test

◦ HTTP 상태 코드

- 특정 HTTP 요청이 성공적으로 완료되었는지를 판별하기 위해 서버에서 응답 상태 코드를 내려줌

- 성공 응답

- 200 OK – 요청이 성공적으로 완료됨

- 클라이언트 에러 응답

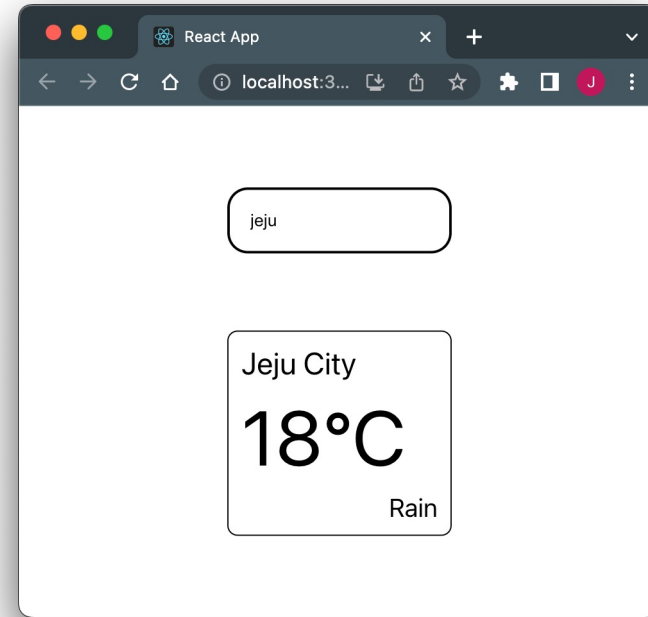
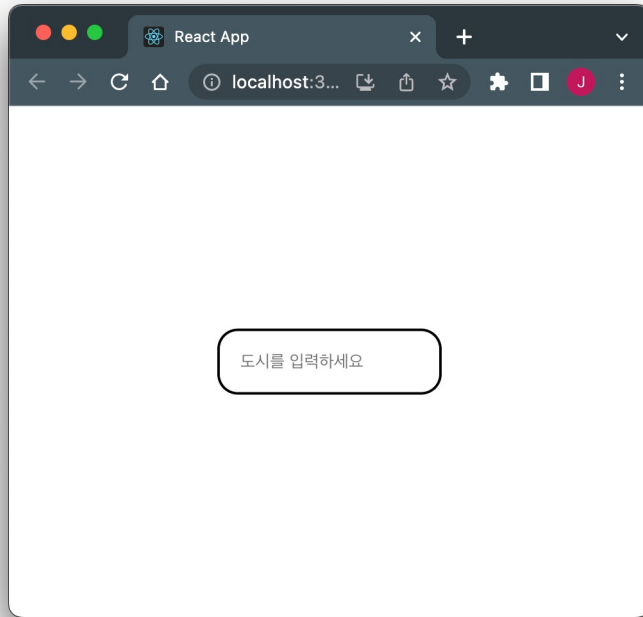
- 400 Bad Request – 잘못된 문법으로 인하여 서버가 요청을 이해할 수 없음
 - 401 Unauthorized – 미승인(비인증)된 요청으로 인한 오류. 인증 토큰 오류 등
 - 403 Forbidden – 콘텐츠에 접근할 권리가 없음
 - 404 Not Found – 서버에서 요청받은 리소스를 찾을 수 없음
 - 405 Method Not Allowed – 지원하지 않는 method로 요청

- 서버 에러 응답

- 500 Internal Server Error – 서버가 처리 방법을 모르는 상황 발생 (적절한 에러처리를 하지 않았을 경우)
 - 503 Service Unavailable – 서버가 요청을 처리할 준비가 되지 않음 (작동 중단, 과부하 등)

PRACTICE 3. 날씨 조회 앱

◦ 실행 결과



PRACTICE 3. 날씨 조회 앱

◦ my-app 프로젝트 – api 연동 모듈

- axios 설치 – npm install axios
- api 연동 모듈 - src/etc/api.js

```
import axios from 'axios';

const BASE_URL = 'https://api.openweathermap.org/data/2.5';
const API_KEY = '5bb4f17b9abe3688a33a27272c989cd7';
```

```
const searchWeather = async (location) => {
  try {
    const data = await axios({
      method: 'GET',
      url: BASE_URL + '/weather',
      params: {
        appid: API_KEY,
        q: location,
        units: 'metric'
      }
    });
    return data;
  } catch(err) {
    alert(err);
    return {};
  }
}

export { searchWeather };
```

PRACTICE 3. 날씨 조회 앱

- UI – src/etc/Weather.jsx

```
import { useState } from 'react';
import styled from 'styled-components';
import { searchWeather } from './api';

function Weather() {
  const [location, setLocation] = useState('');
  const [result, setResult] = useState({});

  const handleKeyDown = async (e) => {
    if (e.key === 'Enter') {
      const data = await searchWeather(location);
      setResult(data);
    }
  }
}
```

```
return (
  <AppWrap>
    <div className="appContentWrap">
      <input
        placeholder="도시를 입력하세요"
        value={location}
        onChange={(e) => setLocation(e.target.value)}
        type="text"
        onKeyDown={handleKeyDown}
      />
      {Object.keys(result).length !== 0 && (
        <ResultWrap>
          <div className="city">{result.data.name}</div>
          <div className="temperature">
            {Math.round(result.data.main.temp * 10) / 10}°C
          </div>
          <div className="sky">{result.data.weather[0].main}</div>
        </ResultWrap>
      )}
    </div>
  </AppWrap>
);

export default Weather;
```

PRACTICE 3. 날씨 조회 앱

- UI styled – src/etc/Weather.jsx

```
const AppWrap = styled.div`
  width: 100vw;
  height: 100vh;

  .appContentWrap {
    left: 50%;
    top: 50%;
    transform: translate(-50%, -50%);
    position: absolute;
    padding: 20px;
  }
  input {
    padding: 16px;
    border: 2px black solid;
    border-radius: 16px;
  }
`;
```

```
const ResultWrap = styled.div`
  margin-top: 60px;
  border: 1px black solid;
  padding: 10px;
  border-radius: 8px;

  .city {
    font-size: 24px;
  }
  .temperature {
    font-size: 60px;
    margin-top: 8px;
  }
  .sky {
    font-size: 20px;
    text-align: right;
    margin-top: 8px;
  }
`;
```