

# JavaScript

◆ API

정수아

# Contents

**01** API란?

**02** 날씨 API

**03** 서울시 공공 API - 공공 자전거 정보

**04** 카카오 지도 API



01

**API란?**

# API란?

## ❖ API(Application Programming Interface)

- 응용 프로그램에서 데이터를 주고 받기 위한 방법
- API 사용 시, 개발 시간 및 비용 절감 가능

## ❖ Open API

- 누구나 사용할 수 있도록 공개된 API
- 지도, 날씨, 음악, 공공데이터, 쇼핑 등 다양한 분야에서 사용 가능
- 제공처
  - 네이버, 카카오, 구글 등



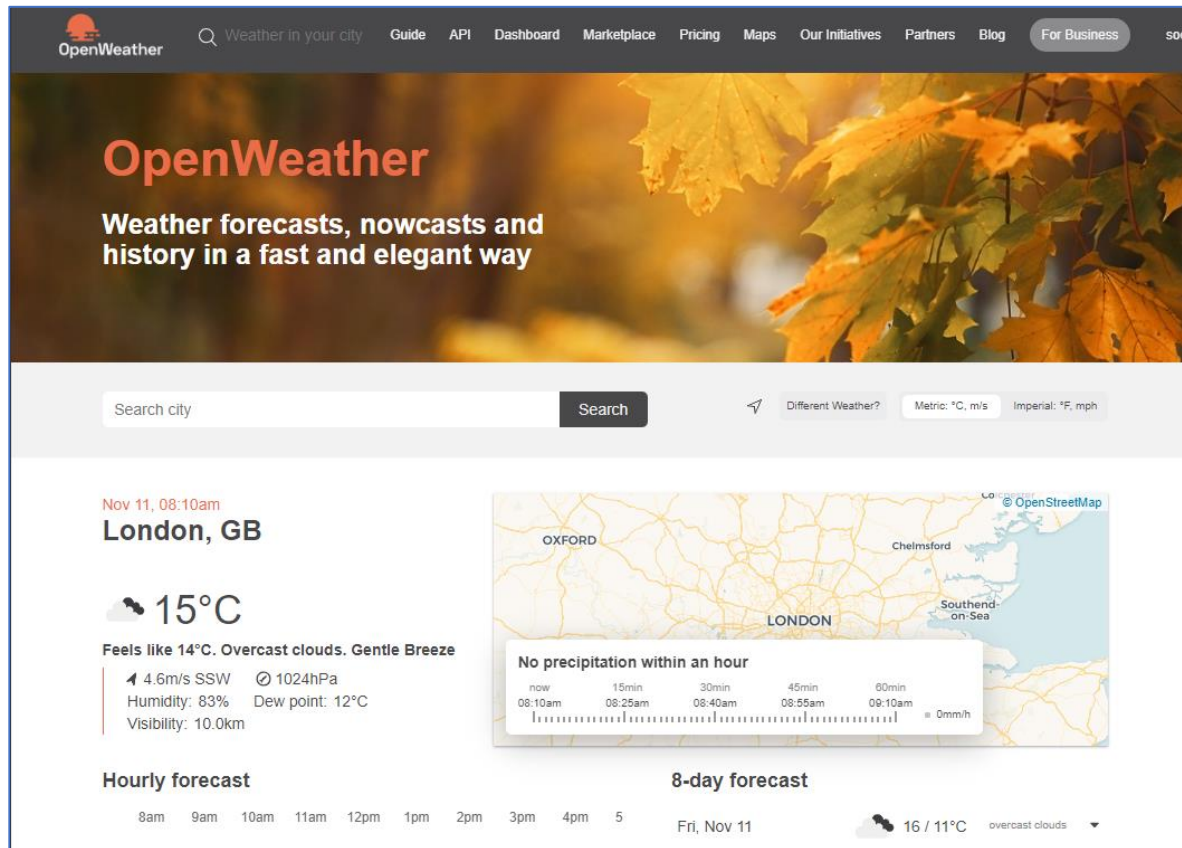
02

**날씨 API**

# 날씨 API

## ❖ 날씨 API

- <https://openweathermap.org/>



## ❖ 회원가입

### Create New Account

Username

Enter email

Password

Repeat Password

We will use information you provided for management and administration purposes, and for keeping you informed by mail, telephone, email and SMS of other products and services from us and our partners. You can proactively manage your preferences or opt-out of communications with us at any time using Privacy Centre. You have the right to access your data held by us or to request your data to be deleted. For full details please see the OpenWeather [Privacy Policy](#).

☐ I am 16 years old and over

☐ I agree with [Privacy Policy](#), [Terms and conditions of sale](#) and [Websites terms and conditions of use](#)


I consent to receive communications from OpenWeather Group of Companies and their partners:

☐ System news (API usage alert, system update, temporary system shutdown, etc)

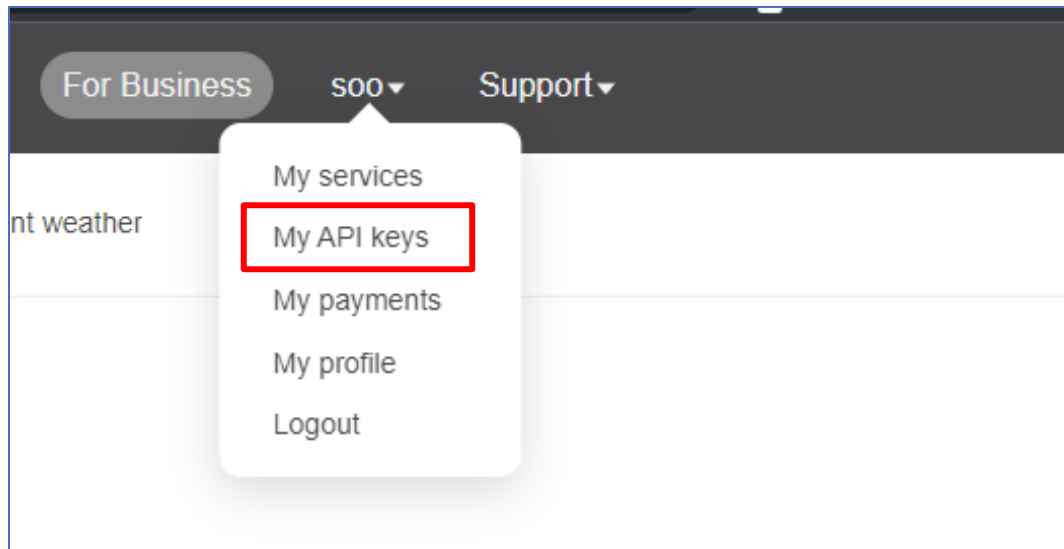
☐ Product news (change to price, new product features, etc)

☐ Corporate news (our life, the launch of a new service, etc)

☐ 로봇이 아닙니다.

  
reCAPTCHA  
개인정보 보호 · 약관

## ❖ 개인 API KEY 발급











## ❖ 개인 API KEY 발급 및 확인

[New Products](#) [Services](#) [API keys](#) [Billing plans](#) [Payments](#) [Block logs](#) [My orders](#) [My profile](#) [Ask a question](#)

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name	Status	Actions
	weather	Active	  
	Default	Active	  

### Create key

Generate

## ❖ 상단 API 메뉴 선택

### Current & Forecast weather data collection

#### Current Weather Data

[API doc](#) [Subscribe](#)

- Access current weather data for any location including over 200,000 cities
- We collect and process weather data from different sources such as global and local weather models, satellites, radars and a vast network of weather stations
- JSON, XML, and HTML formats
- Included in both free and paid subscriptions

#### Hourly Forecast 4 days

[API doc](#) [Subscribe](#)

- Hourly forecast is available for 4 days
- Forecast weather data for 96 timestamps
- JSON and XML formats
- Included in the Developer, Professional and Enterprise subscription plans

#### Daily Forecast 16 days

[API doc](#) [Subscribe](#)

- 16 days forecast is available for any location on the globe
- 1-day step for 16 days
- JSON and XML formats
- Included in all paid subscription plans

#### Climatic Forecast 30 days

[API doc](#) [Subscribe](#)

- Forecast weather data for 30 days
- JSON format
- Included in the Developer, Professional and Enterprise subscription plans

#### Bulk Download

[API doc](#) [Subscribe](#)

- You may request current weather and forecasts in bulk with a variable data granulation
- Current weather bulk is available for 209,000+ cities

#### Global Weather Alerts Push notifications

[Doc](#) [Get access](#)

- Get all the **warnings from national weather agencies**
- Weather alerts are pushed to your endpoint as soon as they occur

## ❖ API 사용 설명 페이지

### Current weather data

Access current weather data for any location on Earth including over 200,000 cities! We collect and process weather data from different sources such as global and local weather models, satellites, radars and a vast network of weather stations. Data is available in JSON, XML, or HTML format.

### Call current weather data

#### How to make an API call

##### API call

```
https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key}
```

##### Parameters

lat, lon	required	Geographical coordinates (latitude, longitude). If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our <a href="#">Geocoding API</a> .
appid	required	Your unique API key (you can always find it on your account page under the "API key" tab)
mode	optional	Response format. Possible values are <code>xml</code> and <code>html</code> . If you don't use the <code>mode</code> parameter format is JSON by default. <a href="#">Learn more</a>
units	optional	Units of measurement. <code>standard</code> , <code>metric</code> and <code>imperial</code> units are available. If you do not use the <code>units</code> parameter, <code>standard</code> units will be applied by default. <a href="#">Learn more</a>

# 날씨 API

## ❖ API call

- 위도, 경도 값으로 날씨 api 호출하기

```
https://api.openweathermap.org/data/2.5/weather?  
lat={lat}&lon={lon}&appid={API key}&units=metric
```

- lat : 위도
- lon : 경도
- API Key : 사용자의 API 키

## ❖ navigator.geolocation

- 웹에서 장치의 위치 정보를 담은 Geolocation 객체를 반환
- .getCurrentPosition(success, error)
  - 장치의 현재 위치를 GeolocationPosition 객체로 반환
  - success
    - 위치 정보 추출 성공 시 실행되는 콜백함수
  - error
    - 위치 정보 추출 실패 시 실행되는 콜백함수

# 위치 정보 얻어오기

## ❖ index.js

```
// API KEY
const API_KEY = "본인의 API KEY";

// 장치의 현재 위치 추출
navigator.geolocation.getCurrentPosition(onGeoLocation, onGeoError);

function onGeoLocation(position) {
  // position : 위치 정보를 담은 객체
  console.log(position);

  const lat = position.coords.latitude; // 위도
  const lon = position.coords.longitude; // 경도
}
```

# 위치 정보 얻어오기

## ❖ position 객체

```
GeolocationPosition {coords: GeolocationCoordinates, timestamp: 1668171846835} ⓘ  
  ▼ coords: GeolocationCoordinates  
    accuracy: 117  
    altitude: null  
    altitudeAccuracy: null  
    heading: null  
    latitude: 37.488778379157345  
    longitude: 126.97115171385907  
    speed: null  
    ▶ [[Prototype]]: GeolocationCoordinates  
    timestamp: 1668171846835  
    ▶ [[Prototype]]: GeolocationPosition
```

# API 호출 및 처리

## ❖ index.js

```
// API 호출 경로
const url = `https://api.openweathermap.org/data/2.5/weather?
            lat=${lat}&lon=${lon}&appid=${API_KEY}&units=metric`;

// URL 처리
fetch(url)
  .then((response) => response.json())
  .then((data) => {
    const weatherImg = document.querySelector("#weather .wImg");
    const weather = document.querySelector("#weather .temp");
    const info = document.querySelector("#weather .info");
    const city = document.querySelector("#weather .city");

    // HTML 태그에 데이터 넣기
    weatherImg.src = "./mist_icon.png";
    weather.innerHTML = `${Math.ceil(data.main.temp)}°C`;
    info.innerHTML = `${data.weather[0].main}`;
    city.innerHTML = data.name;
  });
```



# API 호출 및 처리

## ❖ data 객체

```
▼ {coord: {...}, weather: Array(1), base: 'stations', main: {...}, visibility: 9000, ...} ⓘ
  base: "stations"
  ▶ clouds: {all: 0}
  ▶ cod: 200
  ▶ coord: {lon: 126.9735, lat: 37.4889}
  ▶ dt: 1668173730
  ▶ id: 6800035
  ▼ main:
    feels_like: 9.8
    humidity: 84
    pressure: 1025
    temp: 10.5
    temp_max: 10.87
    temp_min: 9.85
  ▶ [[Prototype]]: Object
  ▶ name: "Banpobondong"
  ▶ sys: {type: 1, id: 8096, country: 'KR', sunrise: 1668118016, sunset: 1668155120}
    timezone: 32400
    visibility: 9000
  ▼ weather: Array(1)
    ▶ 0: {id: 701, main: 'Mist', description: 'mist', icon: '50n'}
      length: 1
    ▶ [[Prototype]]: Array(0)
  ▶ wind: {speed: 1.03, deg: 210}
  ▶ [[Prototype]]: Object
```

위치 정보

날씨 정보

지역 정보

날씨 정보

# API 호출 및 처리

## ❖ index.js

- 위치 정보 추출 실패 시 처리하는 함수

```
function onGeoError() {  
    alert("위치를 찾을 수 없습니다.");  
}
```

# API 호출 및 처리

## ❖ main.html

```
<div id="weather">  
  <img class="wImg" src="">  
  <div class="temp"></div>  
  <div class="info"></div>  
  <div class="city"></d>  
</div>  
<script src="./index.js" />
```

## ❖ 실행 결과





03

## 서울시 공공 API - 공공 자전거 정보

# 서울시 공공 API

## ❖ 서울 열린데이터 광장

- <https://data.seoul.go.kr/index.do>

- 회원가입 후 인증키 발급

서울 열린데이터 광장

공공데이터 통계 서울빅데이터 소식&참여 이용안내

로그인 | **회원가입** | 사이트맵

찾고 싶은 데이터를 입력해 주세요. **Q 검색**

지하철 | 따릉이 | 유동인구 | 인구 | 소득

### 서울 시민생활 데이터

서울 시민생활 데이터 2/5 < || >

#### 서울시민을 위한 공공데이터

열린데이터광장은 서울시 및 자치구, 그 산하기관에서 보유하고 있는 공공데이터를 지속적으로 발굴·개방하고 있습니다.

공공데이터	서비스	시학습데이터
7,073	14,582	15

OpenAPI	5,519	SHEET	6,712
FILE	639	LOD	108
CHART	1,328	MAP	128
LINK	148		

# 서울시 공공 API

## ❖ 회원가입

### 회원가입



HOME > 회원 > 회원가입

#### 일반회원

14세이상 내국인



가입하기

#### 어린이회원

만14세미만 내국인



가입하기

#### 외국인회원



가입하기

#### 기업회원



가입하기

# 서울시 공공 API

## ❖ 인증키 발급



The screenshot shows the Seoul Open Data Portal (서울 열린데이터 광장) website. The top navigation bar includes links for '로그아웃' (Logout), '인증키 신청' (API Key Application), '인증키 관리' (API Key Management), '문의 관리' (Inquiry Management), '활용사례(갤러리) 관리' (Usage Cases (Gallery) Management), and '사이트맵' (Sitemap). Below this, there are links for '공공데이터' (Public Data), '통계' (Statistics), '서울빅데이터' (Seoul Big Data), '소식&참여' (News & Participation), and '이용안내' (Usage Guide). The main content area features a search bar with the placeholder text '찾고 싶은 데이터를 입력해 주세요.' (Enter the data you want to find.) and a 'Q 검색' (Search) button. Below the search bar, there are links for '지하철' (Subway), '따릉이' (Ttarengi), '유동인구' (Floating Population), '인구' (Population), and '소득' (Income).



# 서울시 공공 API

## 인증키 안내



열린데이터광장에서 제공하는 오픈API를 사용하기 위해서는 먼저 인증키를 발급받으셔야 합니다.  
오픈API는 다양한 서비스와 데이터를 좀 더 쉽게 이용할 수 있도록 공개한 개발자를 위한 인터페이스입니다.

### ◆ 인증키 발급

- 열린데이터광장에서 제공하는 오픈API를 사용하기 위해서는 먼저 인증키를 발급 받으셔야 합니다.
- 오픈API는 다양한 서비스와 데이터를 좀 더 쉽게 이용할 수 있도록 공개한 개발자를 위한 인터페이스입니다.

### ◆ 인증키 사용

- 실시간 지하철 오픈API의 경우 하루에 최대 1,000회 요청할 수 있습니다.
- 오픈API를 통해서 1회에 최대 1,000건을 요청할 수 있습니다.  
(1,000건이 넘는 경우 나누어서 호출하세요.)
- 이용 제약 없이 계속 사용하시려면 활용사례(갤러리)에 인증키와 함께 콘텐츠를 등록하세요.

일반 인증키 신청

실시간 지하철 인증키 신청

# 서울시 공공 API

\* 필수 입력항목

* 서비스(사용) 환경	<input checked="" type="radio"/> 웹 사이트 개발 <input type="radio"/> 앱개발 (모바일 솔루션 등) <input type="radio"/> 연구 (논문 등) <input type="radio"/> 기타 참고자료		
* 사용URL (150자 이내)	<input type="text"/>		
* 관리용 대표 이메일 (단체/기업/기관)	<input type="text"/>	@	<input type="text"/> <div>선택 ▼</div>
* 활용용도	<input type="text"/>		
* 내용 (200자 이내)	<div> <input type="text"/> </div> <div>0/200자</div>		

▶ 개인정보 수집·이용 동의

일반 인증키 신청 관리를 위해서 아래와 같이 개인정보를 수집합니다.

1. 개인정보 수집 목적: 열린데이터광장 API 활용 인증키 신청 관리
2. 개인정보 수집 항목: 이메일 주소
3. 보유 및 이용기간: 서비스 종료까지

(필수) 위 개인정보 수집 이용에 동의합니다. ☐ 동의함

인증키 신청

# 서울시 공공 API

## ❖ 인증키 확인

일반 인증키(2)		실시간 지하철 인증키 발급대기현황(0)	예약 인증키 발급 대기현황(0)		
정상	일반인증키		인증키 복사	이용내역	활용사례(갤러리) 등록
정상	일반인증키		인증키 복사	이용내역	활용사례(갤러리) 등록

# 서울시 공공 API

❖ 상단 메뉴 → 공공데이터 → 인기그룹데이터 클릭

로그아웃 | 인증키 신청 | 인증키 관리 | 문의 관리 | 활용사례(갤러리) 관리 | 사이트맵

서울 열린데이터 광장

공공데이터

통계

서울빅데이터

소식&참여

이용안내

공공데이터

인기그룹데이터

새로 개방한 데이터

인기데이터

Linked Open Data

자치구 열린데이터

통계표

통계목록

통계분석

서울의하루

서울통계간행물

서울의 인기통계

서울의 100대통계

서울 생활인구

서울 생활이동

서울 시민생활 데이터

서울 실시간 도시데이터

AI학습데이터

데이터소식

공지사항

FAQ

문의하기

활용사례(갤러리)

데이터 수요창구

열광 뉴스레터

열광소개

Open API 소개

공공데이터 제공신청

통계로 본 서울시

2030세대 **삶**의 모습

[자치구별 2030세대 인구 비율]

서울 30.1%



통계로 본 서울시 2030세대 삶의 모습

3/5 < || >

열린데이터광장은 서울시 및 자치구, 그 산하기관에서 보유하고 있는 공공데이터를 지속적으로 발굴·개방하고 있습니다.

공공데이터

7,073

서비스

14,582

AI학습데이터

15

OpenAPI

5,519

FILE

639

CHART

1,328

LINK

148

SHEET

6,712

LOD







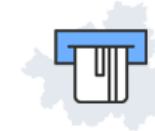
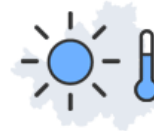




108

MAP

128

# 서울시 공공 API

## ❖ 인기 그룹 데이터

 코로나19	 건설공사정보	 골목상권분석정보	 공공서비스예약정보
 공공자전거이용정보	 공영주차장정보	 교통카드이용정보	 기상관측정보
 기상예보정보	 나눔카지점정보	 모기예보정보	 문화행사정보

# 공공자전거 API

## ❖ 공공자전거 이용 정보

### ◆ 개방데이터 정보

개방데이터 목록	주요 데이터 항목	제공 방식
<a href="#">서울시 공공자전거 대여소 정보 &gt;</a> <a href="#">서울시 공공자전거 이용정보(시간대별) &gt;</a> <a href="#">서울시 공공자전거 이용정보(일별) &gt;</a> <a href="#">서울시 공공자전거 이용정보(월별) &gt;</a> <a href="#">서울시 공공자전거 신규가입자 정보(일별) &gt;</a> <a href="#">서울시 공공자전거 신규가입자 정보(월별) &gt;</a> <a href="#">서울시 공공자전거 대여소별 이용정보(월별) &gt;</a> <a href="#">서울시 공공자전거 이용현황 &gt;</a> <a href="#">서울시 공공자전거 외국인 대여정보(일별) &gt;</a> <a href="#">서울시 공공자전거 외국인 대여정보(월별) &gt;</a> <a href="#">서울시 공공자전거 대여이력 정보 &gt;</a> <a href="#">서울시 공공자전거 실시간 대여정보 &gt;</a>	대여소(구), 대여소ID, 대여소명, 대여소 주소, 위도, 경도, 기준시작 일자, 거치대수, 대여일자, 대여소번호, 대여건수, 반납건수 등	SHEET OpenAPI FILE

# 공공자전거 API

## ❖ 데이터 파일

### 파일내려받기

\* 파일에 이상이 있는 경우 '오류신고'를 통해 운영자에게 알려주세요.

[오류신고](#)

NO	항목	파일명	용량 (MB)	수정일	내려받기
1	데이터	3. 공공자전거 대여소 정보(22.06월 기준).csv	0.26	2022.07.11.	<a href="#">↓</a>
2	데이터	공공자전거 대여소 정보(21.12월 기준).xlsx	0.25	2022.01.20.	<a href="#">↓</a>
3	데이터	공공자전거 대여소 정보(21.06월 기준).xlsx	0.24	2021.07.27.	<a href="#">↓</a>
4	데이터	공공자전거 대여소 정보(21.01.31 기준).csv	0.2	2021.02.22.	<a href="#">↓</a>

대여소 번호	보관소(대여소소재지(위치))	설치 시기	설치형태	운영 방식
	자치구 상세주소	위도	경도	
			LCD	QR
			거치 대수	거치 대수
102	망원역 1번마포구 서울특별시 마포구 월드컵로 72	37.55565	126.9106	15 QR
103	망원역 2번마포구 서울특별시 마포구 월드컵로 79	37.55495	126.9108	14 QR
104	합정역 1번마포구 서울특별시 마포구 양화로 59	37.55063	126.915	13 QR
105	합정역 5번마포구 서울특별시 마포구 양화로 48	37.55001	126.9148	5 QR

# 공공자전거 API

## ❖ 요청 인자

변수명	타입	변수설명	값설명
KEY	String(필수)	인증키	OpenAPI에서 발급된 인증키
TYPE	String(필수)	요청파일타입	xml:xml, xml파일:xmlf, 엑셀파일:xls, json파일:json
SERVICE	String(필수)	서비스명	tbCycleStationInfo
START_INDEX	INTEGER(필수)	요청시작위치	정수 입력 (페이징 시작번호입니다: 데이터 행 시작번호)
END_INDEX	INTEGER(필수)	요청종료위치	정수 입력 (페이징 끝번호입니다: 데이터 행 끝번호)

## ❖ API Call

"http://openapi.seoul.go.kr:8088/개인인증키/json/bikeList/1/5/"



# 공공자전거 API

## ❖ 출력 값

No	출력명	출력설명
공통	list_total_count	총 데이터 건수 (정상조회시 출력됨)
공통	RESULT.CODE	요청결과 코드 (하단 메세지설명 참고)
공통	RESULT.MESSAGE	요청결과 메시지 (하단 메세지설명 참고)
1	STA_LOC	대여소그룹명
2	RENT_ID	대여소ID
3	RENT_NO	대여소번호
4	RENT_NM	대여소명
5	RENT_ID_NM	대여소번호명
6	HOLD_NUM	거치대수
7	STA_ADD1	주소1
8	STA_ADD2	주소2
9	STA_LAT	위도
10	STA_LONG	경도

# [실습] 서울시 공공자전거 정보 가져오기

## ❖ 대여소 정보 5개 호출하기

```
fetch(
  "http://openapi.seoul.go.kr:8088/개인인증키/json/bikeList/1/5/"
)
.then((response) => response.json())
.then((data) => {
  console.log(data);
})
.catch((error) => console.log(error));
```

# [실습] 서울시 공공자전거 정보 가져오기

## ❖ 실행 결과

```
▼ {rentBikeStatus: {...}} ⓘ
  ▼ rentBikeStatus:
    ▶ RESULT: {CODE: 'INFO-000', MESSAGE: '정상 처리되었습니다.'}
    list total count: 5
    ▼ row: Array(5)
      ▶ 0: {rackTotCnt: '15', stationName: '102. 망원역 1번출구 앞', parkingBikeTotCnt: '30', shared: '200', stationLa
      ▶ 1: {rackTotCnt: '14', stationName: '103. 망원역 2번출구 앞', parkingBikeTotCnt: '44', shared: '314', stationLa
      ▶ 2: {rackTotCnt: '13', stationName: '104. 합정역 1번출구 앞', parkingBikeTotCnt: '2', shared: '15', stationLa
      ▶ 3: {rackTotCnt: '5', stationName: '105. 합정역 5번출구 앞', parkingBikeTotCnt: '1', shared: '20', stationLa
      ▶ 4: {rackTotCnt: '12', stationName: '106. 합정역 7번출구 앞', parkingBikeTotCnt: '1', shared: '8', stationLa
    length: 5
    ▶ [[Prototype]]: Array(0)
    ▶ [[Prototype]]: Object
    ▶ [[Prototype]]: Object
```

# [실습] 서울시 공공자전거 정보 가져오기

## ❖ 대여소 정보만 가져오기

```
fetch(
  "http://openapi.seoul.go.kr:8088/개인인증키/json/bikeList/1/5/"
)
.then((response) => response.json())
.then((data) => {
  let rows = data["rentBikeStatus"]["row"];
  console.log(rows);
})
.catch((error) => console.log(error));
```

# [실습] 서울시 공공자전거 정보 가져오기

## ❖ 실행 결과

```
▼ (5) [{...}, {...}, {...}, {...}, {...}] ⓘ  
  ▶ 0: {rackTotCnt: '15', stationName: '102. 망원역 1번출구 앞', parkingBikeTotCnt: '33', shared: '220', stationLat  
  ▶ 1: {rackTotCnt: '14', stationName: '103. 망원역 2번출구 앞', parkingBikeTotCnt: '42', shared: '300', stationLat  
  ▶ 2: {rackTotCnt: '13', stationName: '104. 합정역 1번출구 앞', parkingBikeTotCnt: '3', shared: '23', stationLatit  
  ▶ 3: {rackTotCnt: '5', stationName: '105. 합정역 5번출구 앞', parkingBikeTotCnt: '2', shared: '40', stationLatitu  
  ▶ 4: {rackTotCnt: '12', stationName: '106. 합정역 7번출구 앞', parkingBikeTotCnt: '1', shared: '8', stationLatitu  
    length: 5  
  ▶ [[Prototype]]: Array(0)
```

# [실습] 서울시 공공자전거 정보 가져오기

## ❖ 대여소 이름(stationName)만 가져오기

```
fetch(
  "http://openapi.seoul.go.kr:8088/개인인증키/json/bikeList/1/5/"
)
.then((response) => response.json())
.then((data) => {
  let rows = data["rentBikeStatus"]["row"];

  rows.forEach((row) => {
    let stationName = row["stationName"];
    console.log(stationName);
  });
})
.catch((error) => console.log(error));
```

# [실습] 서울시 공공자전거 정보 가져오기

## ❖ 실행 결과

102.	망원역	1번출구	앞
103.	망원역	2번출구	앞
104.	합정역	1번출구	앞
105.	합정역	5번출구	앞
106.	합정역	7번출구	앞



04

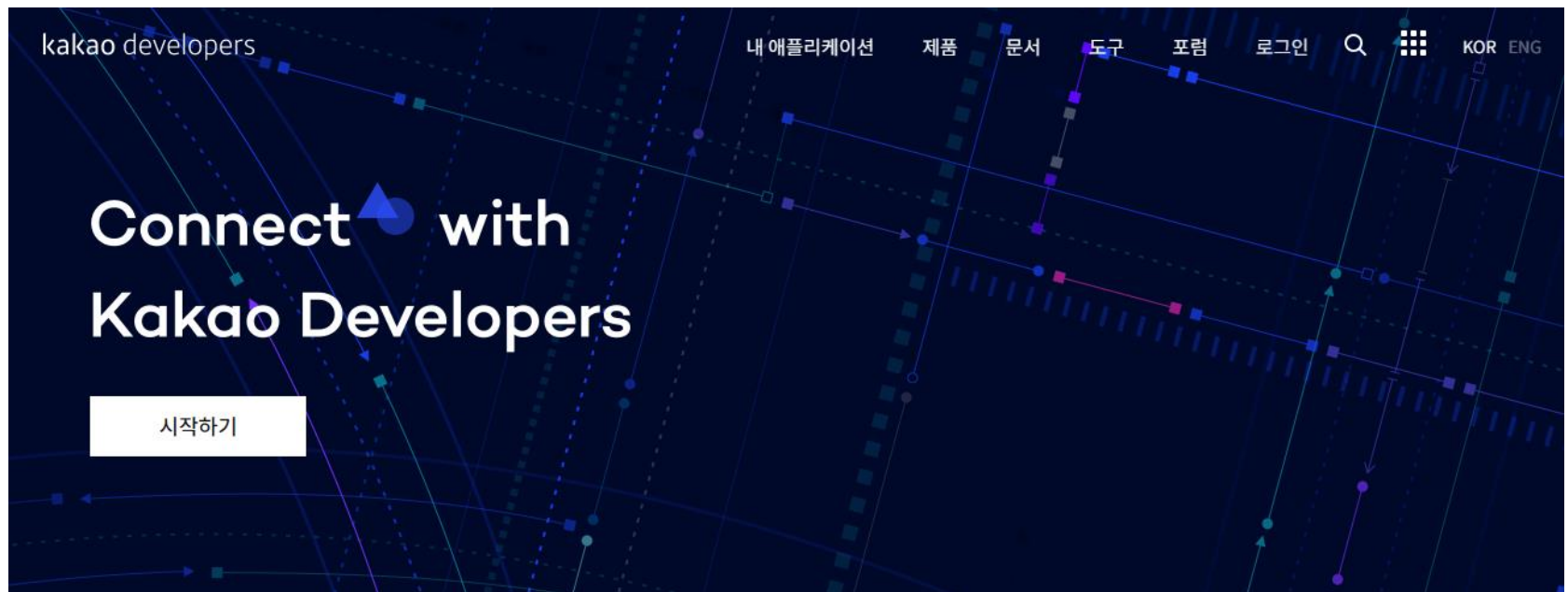
## 카카오 지도 API



# 카카오 API

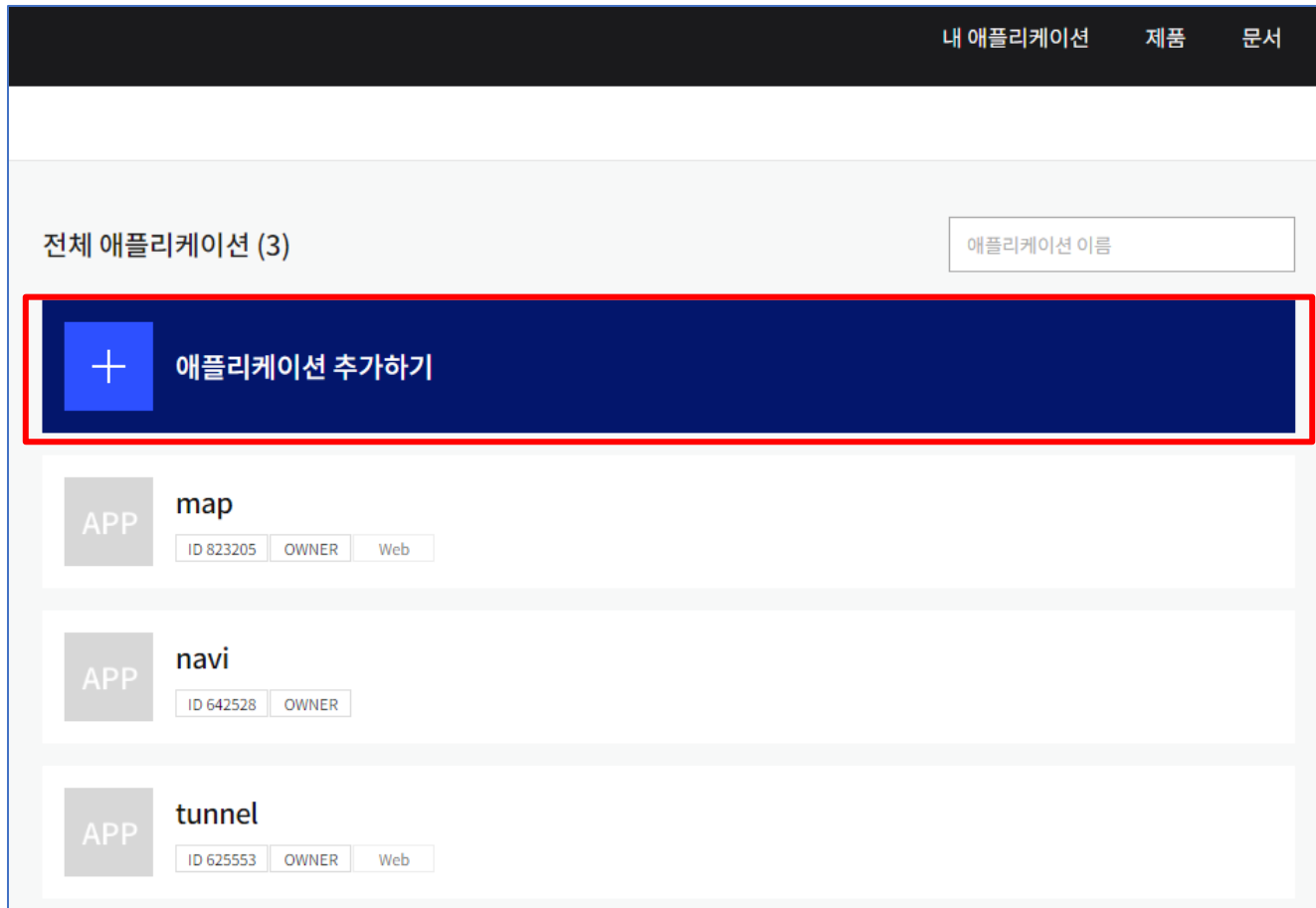
## ❖ 카카오 개발자센터

- <https://developers.kakao.com/>



# 카카오 API

❖ 상단 메뉴 → 내 애플리케이션 → 애플리케이션 추가하기



# 카카오 API

## ❖ 애플리케이션 추가하기

### 애플리케이션 추가하기

앱 아이콘

이미지  
업로드

파일 선택

JPG, GIF, PNG  
권장 사이즈 128px, 최대 250KB

앱 이름

cycleMap

사업자명

soo

- 입력된 정보는 사용자가 카카오 로그인을 할 때 표시됩니다.
- 정보가 정확하지 않은 경우 서비스 이용이 제한될 수 있습니다.

☒

서비스 이용이 제한되는 카테고리, 금지된 내용, 금지된 행동 관련 운영정책을 위반하지 않는 앱입니다.

취소

저장

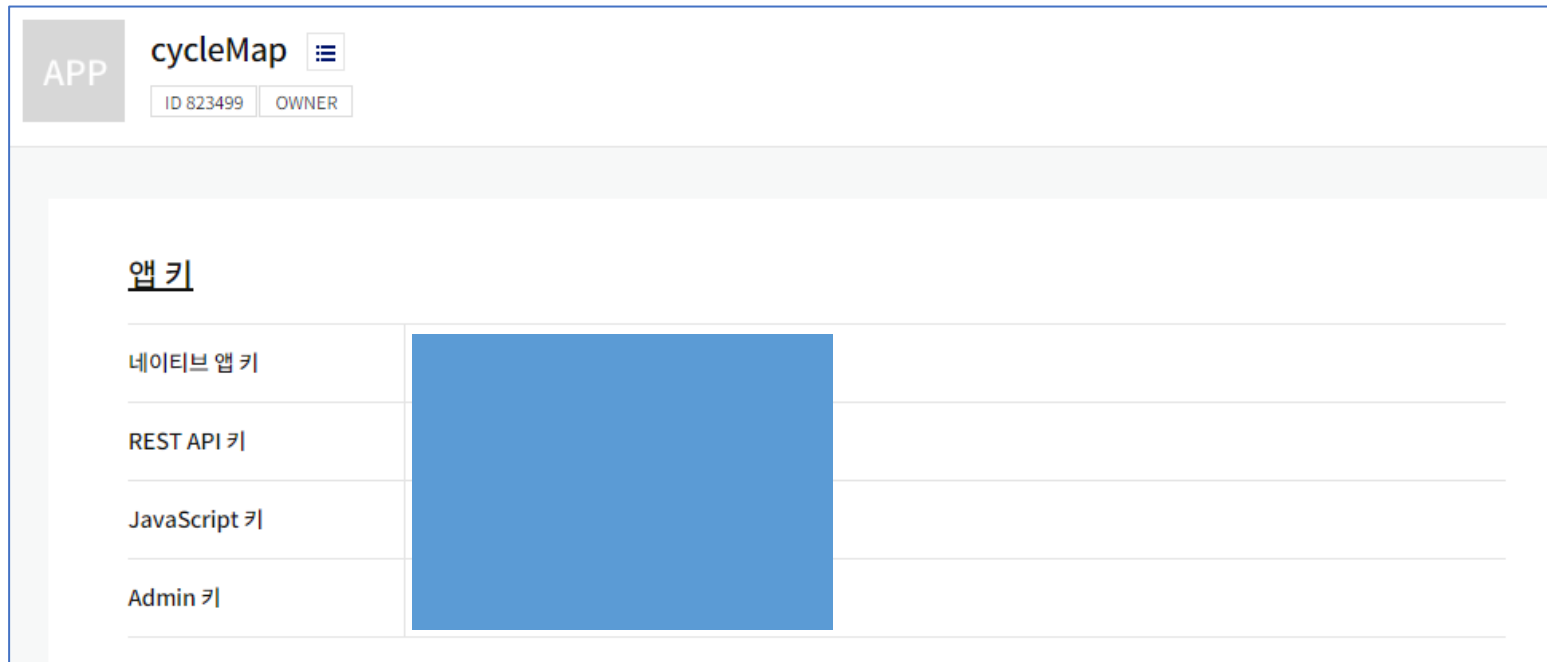
# 카카오 API

## ❖ 애플리케이션 목록

전체 애플리케이션 (4)		애플리케이션 이름
<div><div>+</div>애플리케이션 추가하기</div>		
APP	cycleMap	ID 823499 OWNER
APP	map	ID 823205 OWNER Web
APP	navi	ID 642528 OWNER
APP	tunnel	ID 625553 OWNER Web

# 카카오 API

## ❖ API Key 확인



# 카카오 API

## ❖ 플랫폼 설정하기

플랫폼

설정된 플랫폼 정보가 없습니다. [플랫폼 설정하기](#)



Web

Web 플랫폼 등록

# 카카오 API

## ❖ Web 플랫폼 등록

### Web 플랫폼 등록

#### 사이트 도메인

JavaScript SDK, 카카오톡 공유, 카카오톡, 메시지 API 사용시 등록이 필요합니다.

여러개의 도메인은 줄바꿈으로 추가해주세요. 최대 10까지 등록 가능합니다. 추가 등록은 포럼(데브톡)으로 문의주세요.

예시: (O) <https://example.com> (X) <https://www.example.com>

<http://127.0.0.1:5500>

#### 기본 도메인

기본 도메인은 첫 번째 사이트 도메인으로, 카카오톡 공유와 카카오톡 메시지 API를 통해 발송되는 메시지의 Web 링크 기본값으로 사용됩니다.

<http://127.0.0.1:5500>

취소

저장

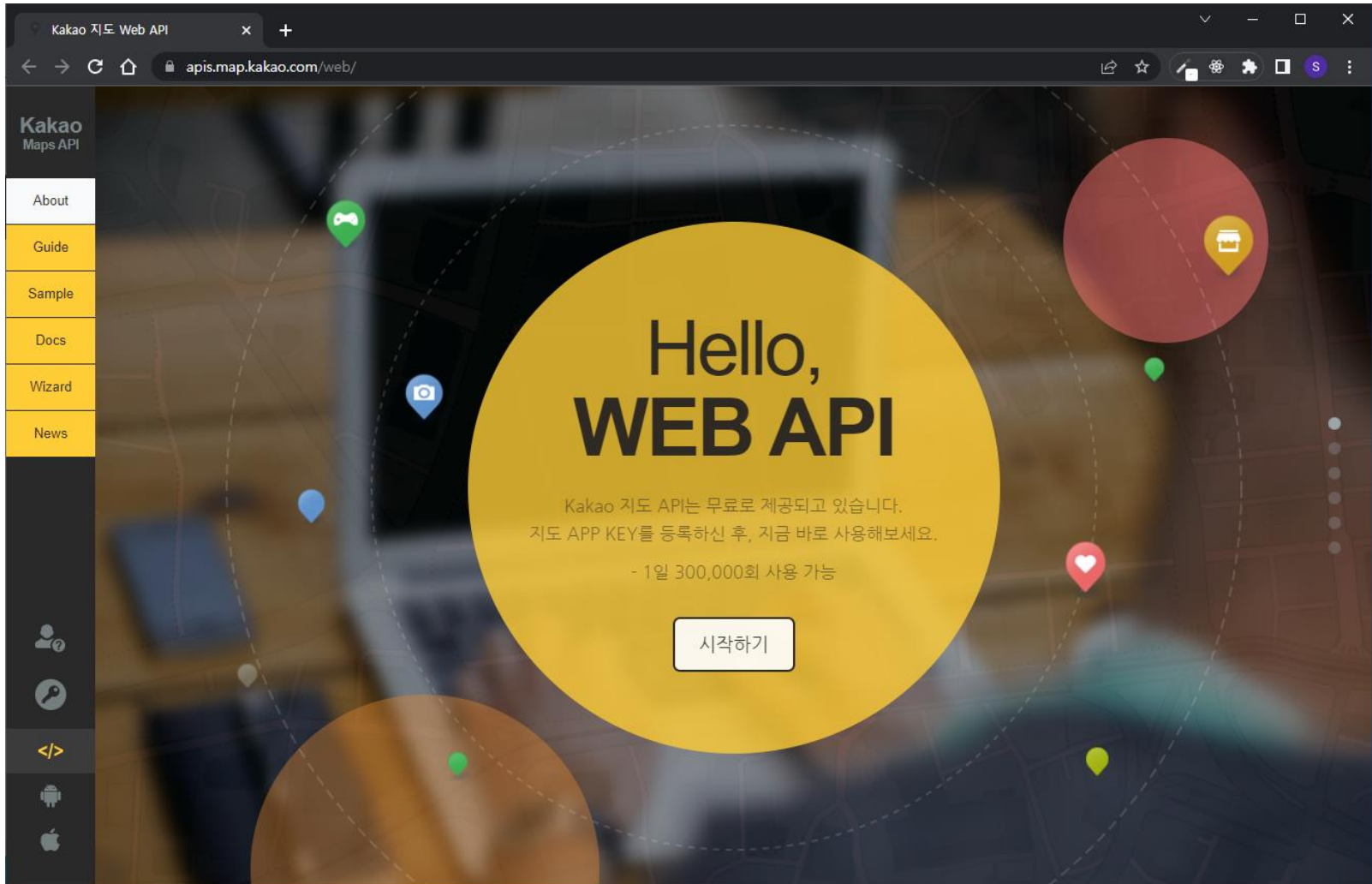
# 카카오 API

❖ 상단 메뉴 → 문서 → 지도/로컬 API 가이드

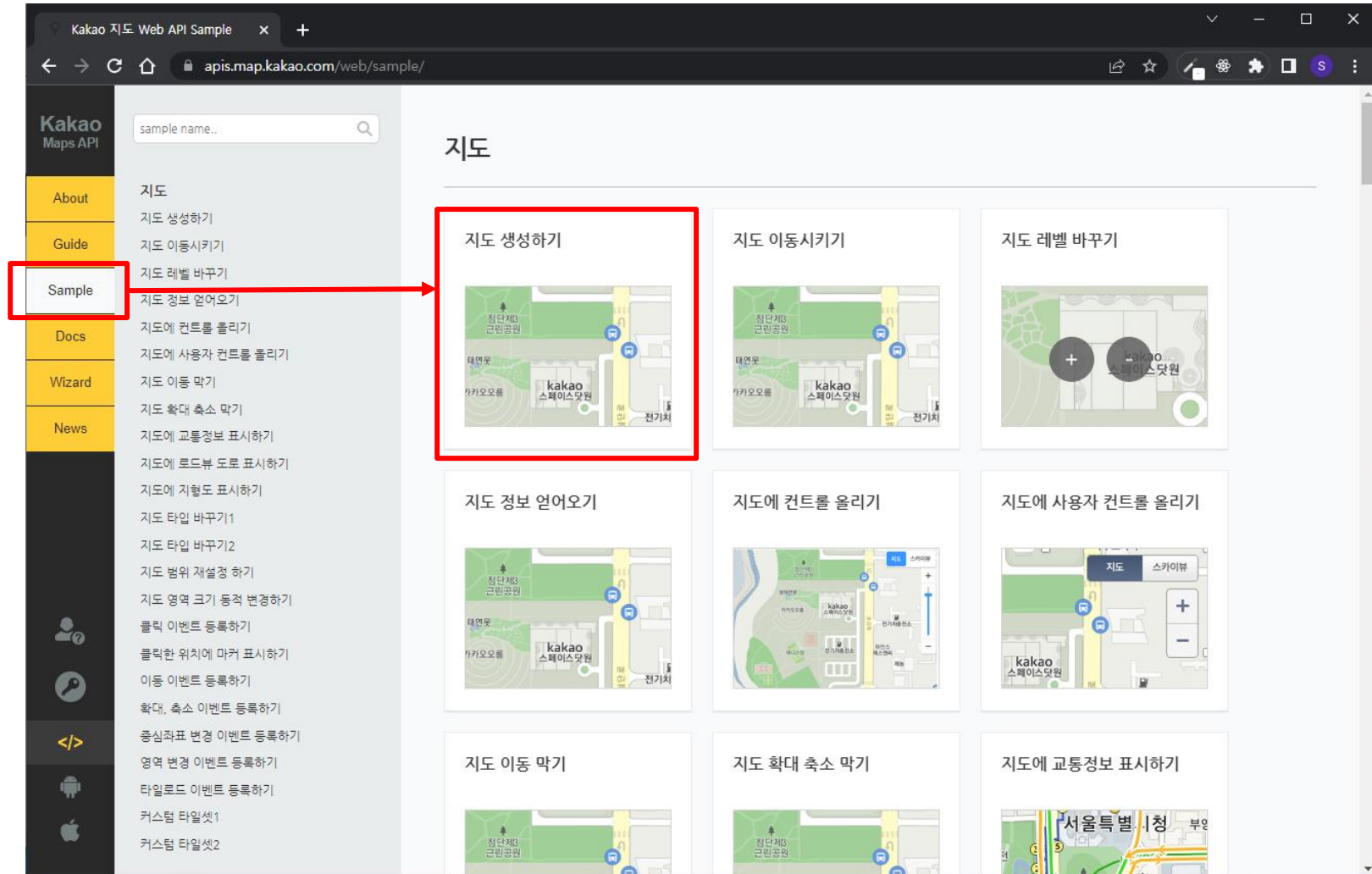




# 카카오 지도 API



# 카카오 지도 API



# 카카오 지도 API

## 지도 생성하기

지도를 생성하는 가장 기본적인 예제입니다.



JavaScript

JavaScript + HTML

직접 해보기

```
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
        center: new kakao.maps.LatLng(33.450701, 126.570667), // 지도의 중심좌표
        level: 3 // 지도의 확대 레벨
    };

// 지도를 표시할 div와 지도 옵션으로 지도를 생성합니다
var map = new kakao.maps.Map(mapContainer, mapOption);
```

# [실습] 카카오 지도 띄우기

## ❖ HTML 파일

- 지도 출력할 영역 생성
- 카카오 지도 API 연결

```
<div id="map" style="width: 100%; height: 500px"></div>

<script type="text/javascript"
      src="//dapi.kakao.com/v2/maps/sdk.js?appkey=개인인증키">
</script>
```

# [실습] 카카오 지도 띄우기

## ❖ JS 파일

```
// 지도 표시 영역
let mapContainer = document.getElementById("map");

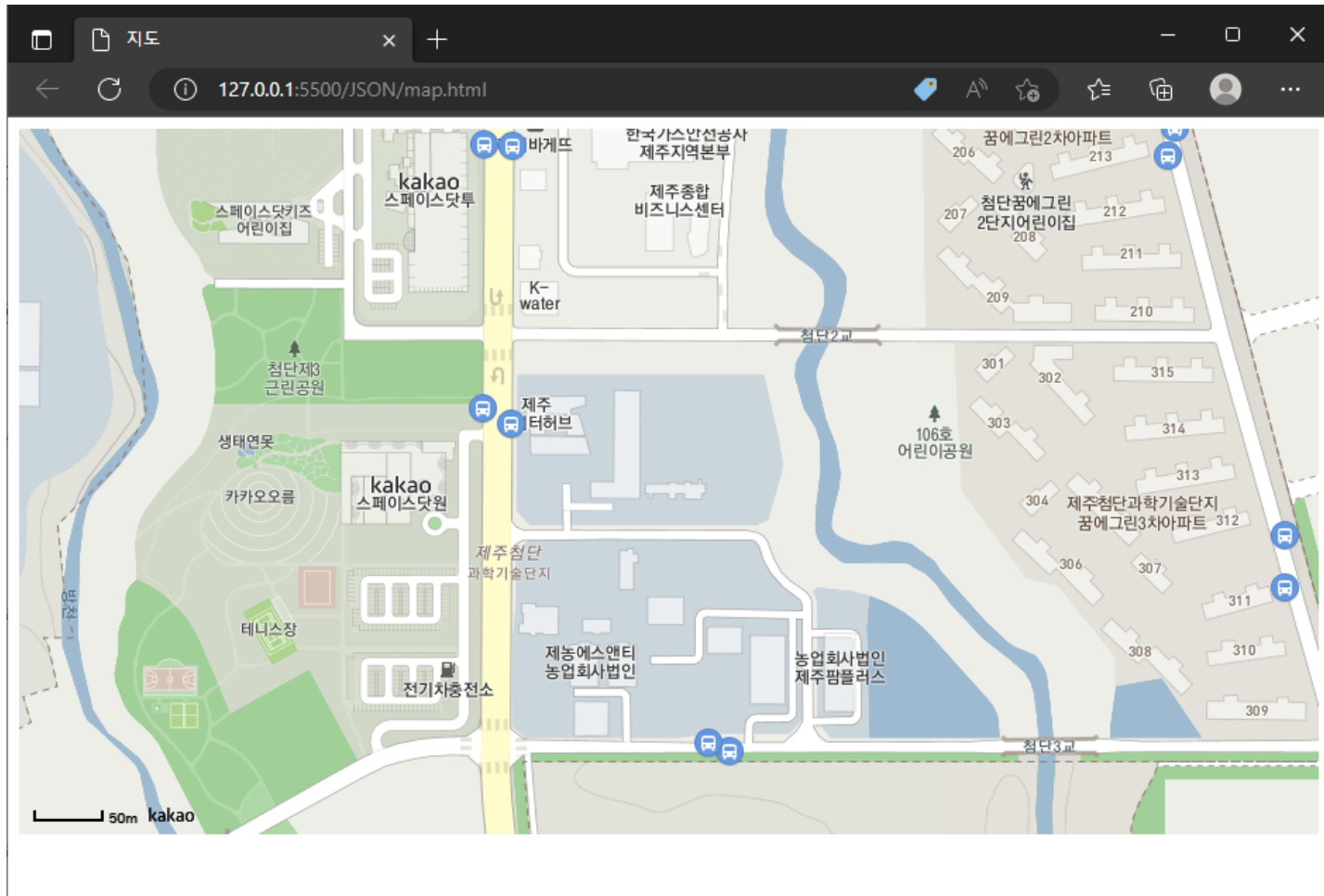
// 지도 옵션
let mapOption = {
  // 지도 중심좌표
  center: new kakao.maps.LatLng(33.450701, 126.570667),

  // 지도 확대 레벨
  level: 3
};

// 지도를 표시할 div와 지도 옵션으로 지도를 생성
let map = new kakao.maps.Map(mapContainer, mapOption);
```

# [실습] 카카오 지도 띄우기

## ❖ 실행 결과



# [실습] 서울시 공공자전거 대여소 지도에 표시

## ❖ 서울시 API를 호출하여 공공자전거 대여소 정보 얻기

```
let stationList = [];  
  
fetch(  
  http://openapi.seoul.go.kr:8088/개인인증키/json/bikeList/1/5/  
).then((response) => response.json())  
  .then(data => {  
    let rows = data["rentBikeStatus"]["row"];  
  
    // 정거장, 위도/경도 저장  
    rows.forEach((row) => {  
      stationName = row["stationName"];  
      stationLatitude = row["stationLatitude"];  
      stationLongitude = row["stationLongitude"];  
  
      // 정거장 정보(stationInfo) 객체 생성  
      stationInfo = {  
        stationName: stationName,  
        stationLatitude: stationLatitude, // 위도  
        stationLongitude: stationLongitude, // 경도  
      };  
  
      // 정거장 정보(stationInfo)를 stationList 배열에 추가  
      stationList.push(stationInfo);  
    });  
  });
```

# [실습] 서울시 공공자전거 대여소 지도에 표시

## ❖ 카카오 지도 API 포맷으로 위치 정보 저장

```
fetch(  
  http://openapi.seoul.go.kr:8088/개인인증키/json/bikeList/1/5/  
).then((response) => response.json())  
.then(data => {  
  // 생략  
  
  // 위치 정보를 저장할 positions 배열  
  let positions = [];  
  
  // 위치 정보를 저장  
  stationList.forEach((info) => {  
    positions.push({  
      latlng: new kakao.maps.LatLng(  
        info.stationLatitude,  
        info.stationLongitude  
      ),  
      title: info.stationName,  
    });  
  });  
  
  // 지도 출력 기능을 가진 함수 호출  
  showMap(positions);  
});
```



# [실습] 서울시 공공자전거 대여소 지도에 표시

## ❖ 지도 출력 기능 함수 구현

```
function showMap(positions) {  
  // 지도 표시 영역  
  let mapContainer = document.getElementById("map");  
  
  // 지도 옵션  
  let mapOption = {  
    center: new kakao.maps.LatLng(37.5556488, 126.91062927),  
    level: 3,  
    marker: positions,  
  };  
  
  // 지도를 표시할 div와 지도 옵션으로 지도를 생성  
  let map = new kakao.maps.Map(mapContainer, mapOption);  
}
```

# [실습] 서울시 공공자전거 대여소 지도에 표시

## ❖ 지도 출력 기능 함수 구현

```
function showMap(positions) {  
  // 생략  
  
  // 이미지 마커 경로  
  let imageSrc =  
    https://t1.daumcdn.net/localimg/localimages/07/mapapidoc/markerStar.png;  
  
  // 이미지 마커 표시  
  positions.forEach((item) => {  
    let imageSize = new kakao.maps.Size(24, 35);  
    let markerImage = new kakao.maps.MarkerImage(imageSrc, imageSize);  
  
    // 마커 생성  
    let marker = new kakao.maps.Marker({  
      map: map,           // 마커를 표시할 지도  
      position: item.latlng, // 마커를 표시할 위치  
      title: item.title,   // 마커 타이틀  
      image: markerImage,  // 마커 이미지  
    });  
  });  
};  
}
```

## ❖ 실행 결과



**THANK 😊 YOU**