

## ■ Rest API 활용 프로그램 만들기

### ● 알고 있어야 하는 내용

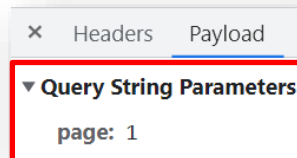
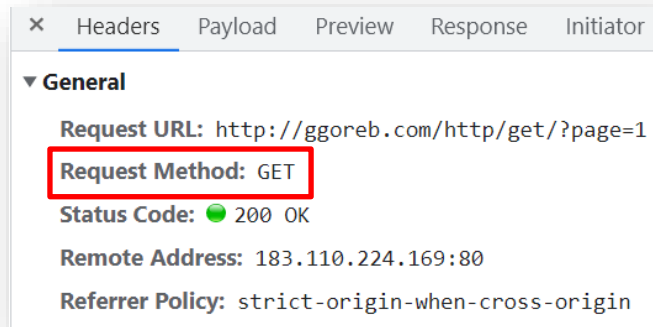
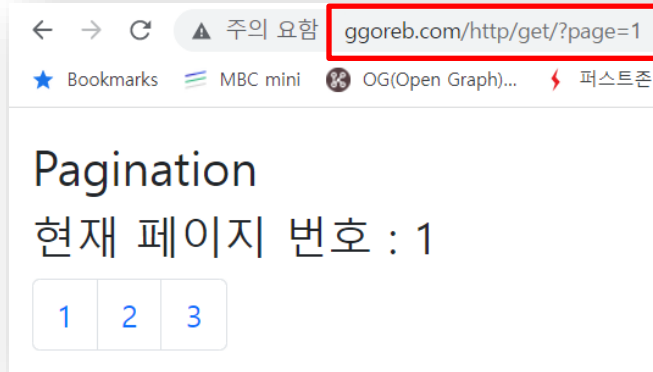
\* **jackson-annotations, jackson-core, jackson-databind** 파일 다운로드

1. HTTP 메소드 GET / POST
2. HTTP Header
3. 카카오 개발자 사이트 가입 / 로그인 / API 문서
4. 웹 브라우저 확장 프로그램 설치 (Json Formatter / API Tester)
5. ObjectMapper 라이브러리 사용법

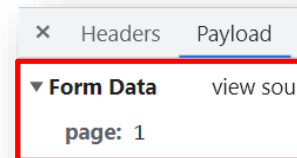
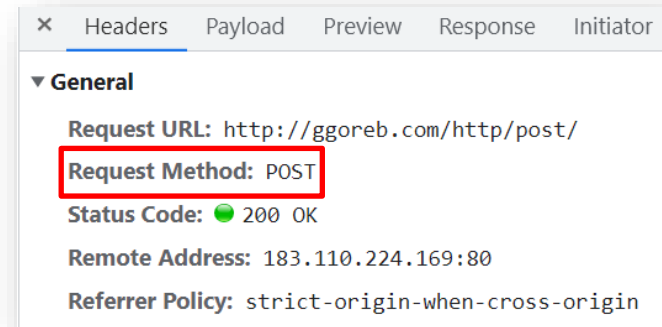
# ■ Rest API 활용 프로그램 만들기

## 1. HTTP 메소드 GET / POST

### GET



### POST



# ■ Rest API 활용 프로그램 만들기

## 1. HTTP 메소드 GET / POST

```
String apiURL = "http://ggoreb.com/http/get/";
apiURL += "?page=1";

URL url = new URL(apiURL);
URLConnection con = (URLConnection) url.openConnection();

BufferedReader br = new BufferedReader(
    new InputStreamReader(con.getInputStream(), "utf-8"));

while(true) {
    String data = br.readLine();
    if(data == null) break;
    System.out.println(data);
}
```

```
<html>
<head>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/boo
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/boo
</head>
<body>
  <div class="container mt-3">
    <h2>Pagination</h2>
    <h3>현재 페이지 번호 : 1</h3>
    <ul class="pagination">
      <li class="page-item"><a class="page-link" href="./?page=1">1</a>
      <li class="page-item"><a class="page-link" href="./?page=2">2</a>
      <li class="page-item"><a class="page-link" href="./?page=3">3</a>
    </ul>
  </div>
</body>
</html>
```

## ■ Rest API 활용 프로그램 만들기

### 1. HTTP 메소드 GET / POST

```
String apiURL = "http://ggoreb.com/http/post/";

URL url = new URL(apiURL);
URLConnection con = (URLConnection) url.openConnection();

/* POST 설정 */
con.setRequestMethod("POST");
con.setDoOutput(true);
con.setDoInput(true);

OutputStream outputStream = con.getOutputStream();
PrintWriter writer = new PrintWriter(
    new OutputStreamWriter(outputStream, "UTF-8"), true);
writer.append("&page=3");
writer.flush();

BufferedReader br = new BufferedReader(
    new InputStreamReader(con.getInputStream(), "utf-8"));

while(true) {
    String data = br.readLine();
    if(data == null) break;
    System.out.println(data);
}
```

## ■ Rest API 활용 프로그램 만들기

### 2. HTTP Header - 헤더값 지정 X

```
String apiURL = "http://ggoreb.com/http/request.jsp";
```

```
URL url = new URL(apiURL);  
HttpURLConnection con = (HttpURLConnection)  
url.openConnection();
```

```
method : GET<br>  
query string<br>  
null  
<br><br>  
header<br>  
key : accept, value : text/html, image/gif, image/jpeg, *, q=.2, */*; q=.  
key : connection, value : close<br>  
key : host, value : ggoreb.com<br>  
key : HOSTING_CONTINENT_CODE, value : AS<br>  
key : HOSTING_COUNTRY_CODE, value : KR<br>  
key : HOSTING_WHITE_IP, value : false<br>  
key : user-agent, value : Java/17.0.2<br>  
key : X-Forwarded-Proto, value : http<br>  
key : X-SERVER_PORT, value : 80<br>  
key : X-SERVER_PROTOCOL, value : HTTP/1.1<br>  
key : X-SIMPLEXI, value : 59.11.183.94<br>  
key : content-length, value : 0<br>
```

## ■ Rest API 활용 프로그램 만들기

### 2. HTTP Header - 헤더값 지정 ○

```
String apiURL = "http://ggoreb.com/http/request.jsp";

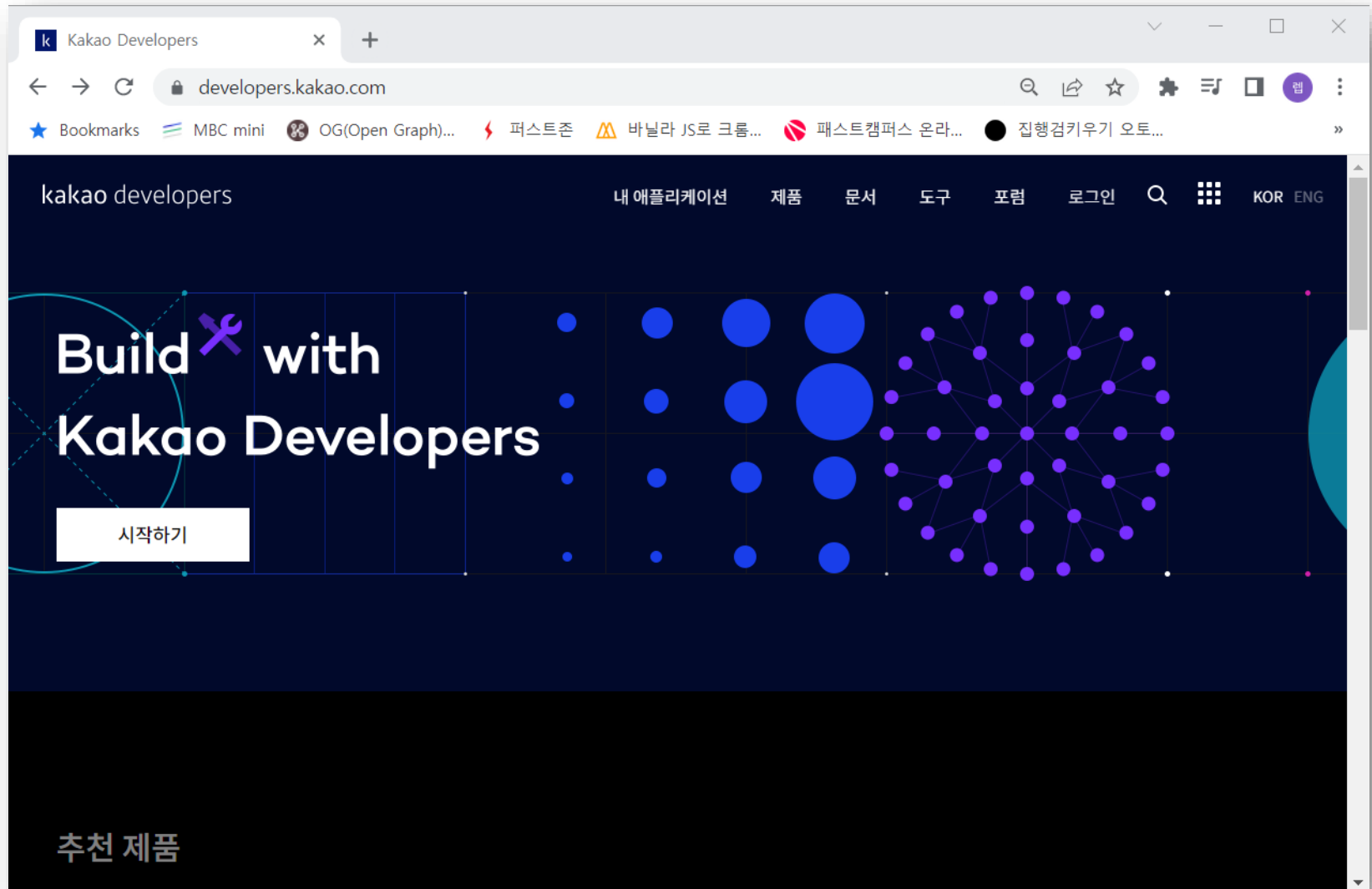
URL url = new URL(apiURL);
URLConnection con = (URLConnection)
url.openConnection();

con.setRequestProperty("accept-language", "ko-kr");
con.setRequestProperty("user-agent", "chrome android");
```

```
method : GET<br>
query string<br>
null
<br><br>
header<br>
key : accept, value : text/html, image/gif, image/jpeg, *; q=.2, */*; q=.
key : accept-language, value : ko-kr<br>
key : connection, value : close<br>
key : host, value : ggoreb.com<br>
key : HOSTING_CONTINENT_CODE, value : AS<br>
key : HOSTING_COUNTRY_CODE, value : KR<br>
key : HOSTING_WHITE_IP, value : false<br>
key : user-agent, value : chrome android<br>
key : X-Forwarded-Proto, value : http<br>
key : X-SERVER_PORT, value : 80<br>
key : X-SERVER_PROTOCOL, value : HTTP/1.1<br>
key : X-SIMPLEXI, value : 59.11.183.94<br>
key : content-length, value : 0<br>
```

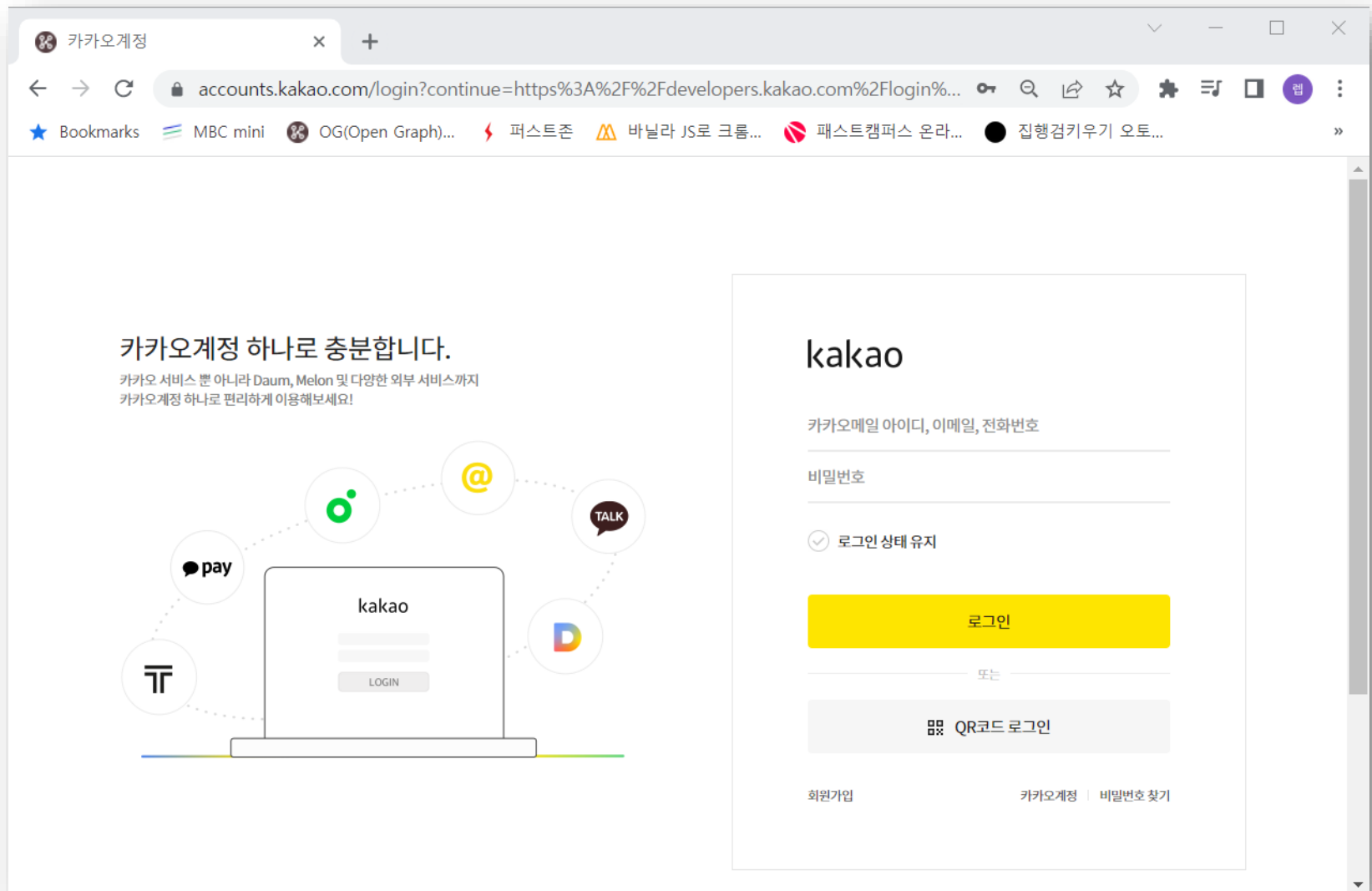
## ■ Rest API 활용 프로그램 만들기

### 3. 카카오 개발자 사이트 가입 / 로그인 / API 문서



## ■ Rest API 활용 프로그램 만들기

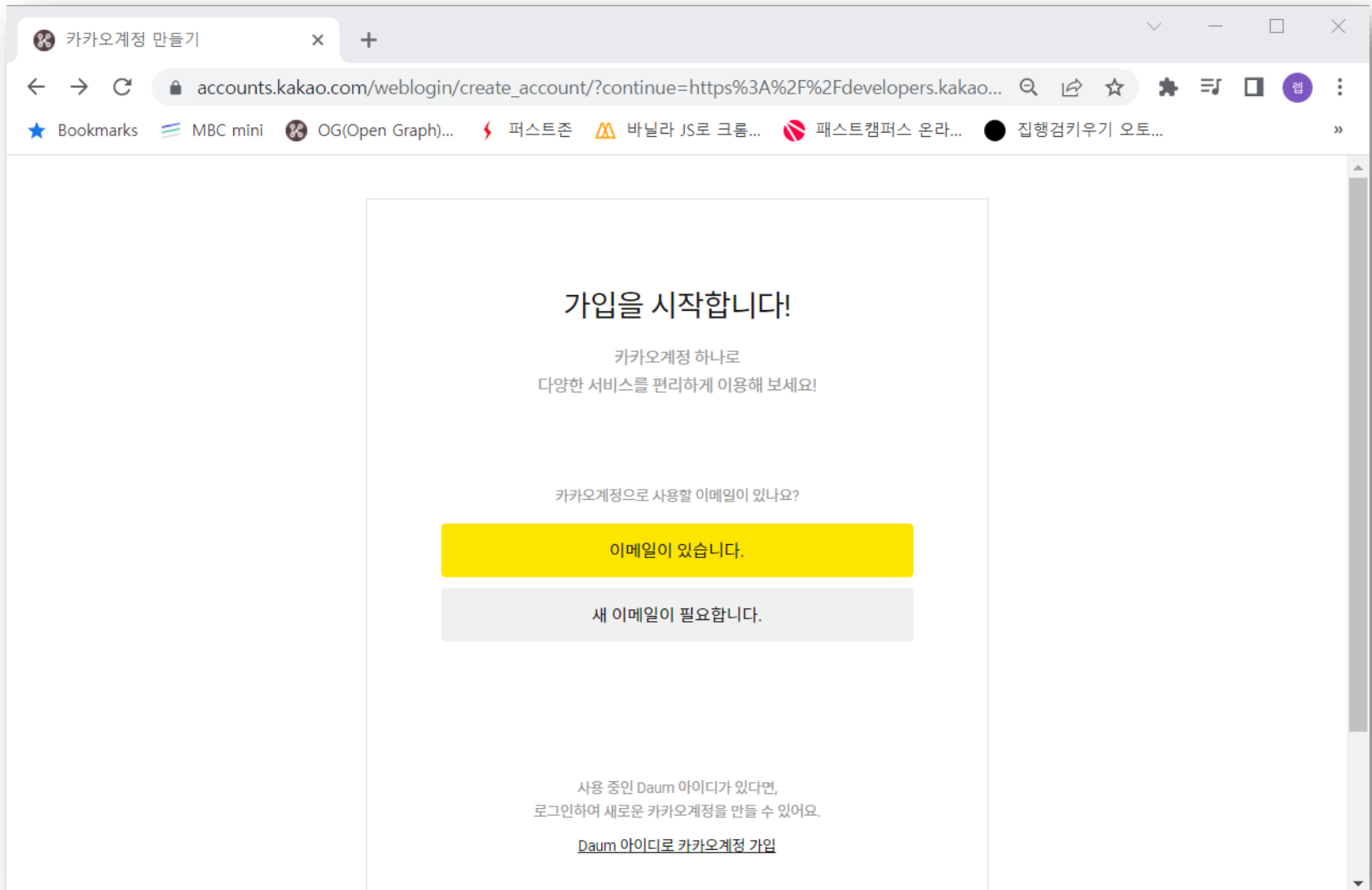
### 3. 카카오 개발자 사이트 가입 / 로그인 / API 문서





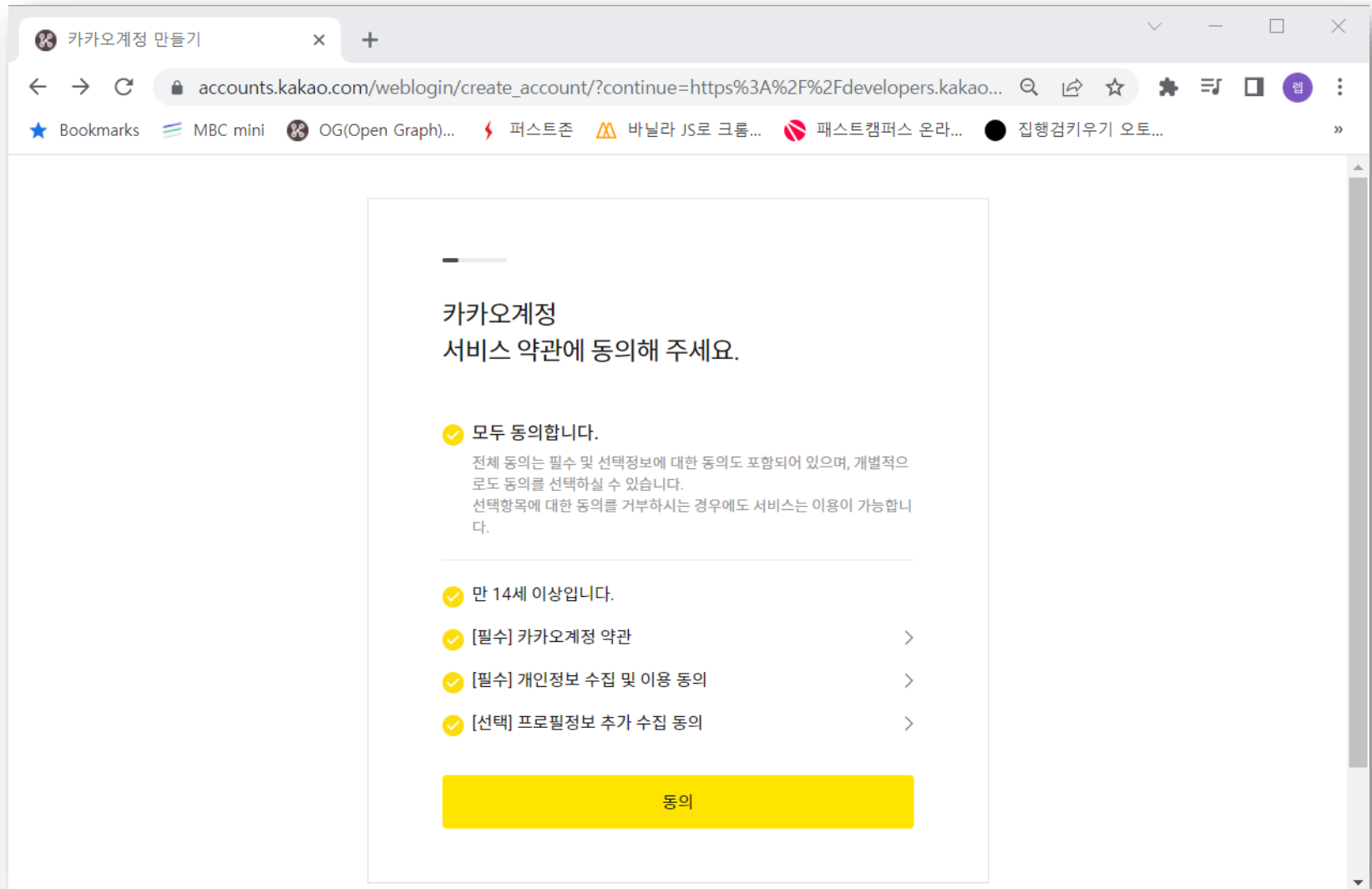
## ■ Rest API 활용 프로그램 만들기

### 3. 카카오 개발자 사이트 가입 / 로그인 / API 문서



## ■ Rest API 활용 프로그램 만들기

### 3. 카카오 개발자 사이트 가입 / 로그인 / API 문서



## ■ Rest API 활용 프로그램 만들기

### 3. 카카오 개발자 사이트 가입 / 로그인 / API 문서

카카오계정 만들기

accounts.kakao.com/weblogin/create\_account/?continue=https%3A%2F%2Fdevelopers.kakao.com...

카카오계정으로 사용할  
이메일 주소를 입력해 주세요.

seorab@naver.com 인증요청

- 입력한 이메일 주소로 인증번호가 발송됩니다.
- 인증메일을 받을 수 있도록 자주 쓰는 이메일 주소를 입력해 주세요.
- 사용 중인 Daum 이메일이 있다면, Daum 아이디로 가입해 주세요.

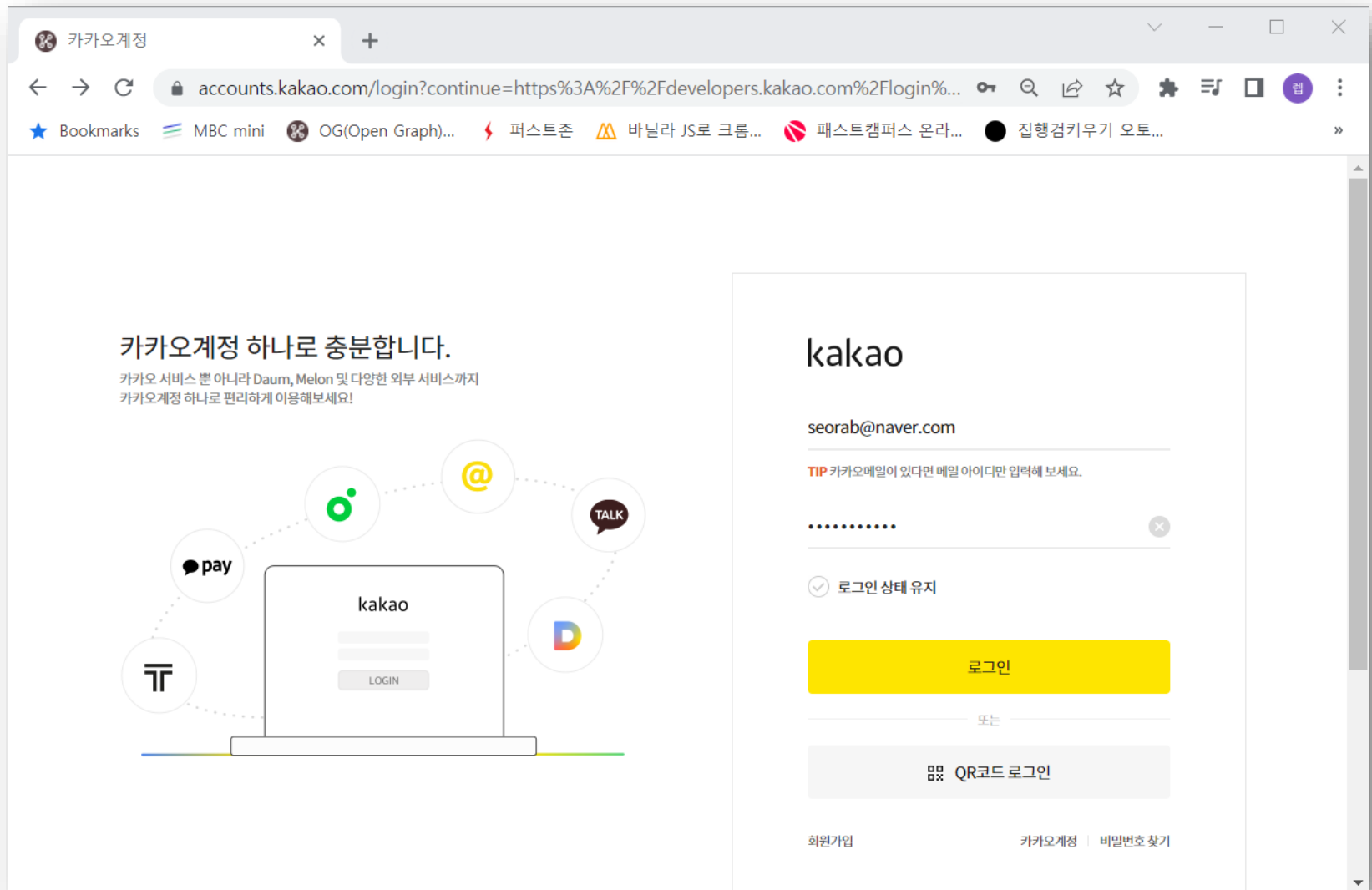
[Daum 아이디로 가입](#)

다음

이용약관 개인정보 처리방침 운영정책 고객센터 공지사항 한국어 ^  
Copyright © Kakao Corp. All rights reserved.

## ■ Rest API 활용 프로그램 만들기

### 3. 카카오 개발자 사이트 가입 / 로그인 / API 문서



## ■ Rest API 활용 프로그램 만들기

### 3. 카카오 개발자 사이트 가입 / 로그인 / API 문서

내 애플리케이션

Kakao Developers 내 애플리케이션

developers.kakao.com/console/app

kakao developers 내 애플리케이션 제품 문서 도구 포럼 seorab@kakao.com KOR ENG

내 애플리케이션

전체 애플리케이션 (3)

애플리케이션 이름

+ 애플리케이션 추가하기

APP	NAME	ID	ROLE	PLATFORMS
APP	API 샘플	ID 434802	OWNER	Web
APP	재난소득가맹점	ID 432597	OWNER	Web
APP	맛있는부산	ID 24827	EDITOR	Android, iOS, Web

## ■ Rest API 활용 프로그램 만들기

### 3. 카카오 개발자 사이트 가입 / 로그인 / API 문서

내 애플리케이션

The screenshot shows the Kakao Developers console interface. A modal window titled '애플리케이션 추가하기' (Add Application) is open in the center. The modal contains the following fields and options:

- 앱 아이콘 (App Icon):** Includes an '이미지 업로드' (Image Upload) button and a '파일 선택' (File Select) button. Below the buttons, it specifies 'JPG, GIF, PNG' and '권장 사이즈 128px, 최대 250KB'.
- 앱 이름 (App Name):** A text input field containing 'API 연습'.
- 사업자명 (Business Name):** A text input field containing '아무거나'.

Below the input fields, there are two bullet points:

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

At the bottom of the modal, there are two buttons: '취소' (Cancel) and '저장' (Save).

The background shows the Kakao Developers console with a sidebar on the left and a top navigation bar. The top bar includes the text 'kakao developers' and navigation links for '내 애플리케이션', '제품', '문서', '도구', and '포럼'. The user's email 'seorab@kakao.com' is also visible.

## ■ Rest API 활용 프로그램 만들기

### 3. 카카오 개발자 사이트 가입 / 로그인 / API 문서

내 애플리케이션

The screenshot shows the Kakao Developers console interface. The browser address bar displays `developers.kakao.com/console/app/759787`. The page title is "Kakao Developers 내 애플리케이션". The main navigation bar includes "내 애플리케이션", "제품", "문서", "도구", "포럼", and a user profile "seorab@kakao.com". The breadcrumb trail is "내 애플리케이션 > 앱 설정 > 요약 정보". The left sidebar shows "APP" and "API 연습". The main content area displays a table of API keys:

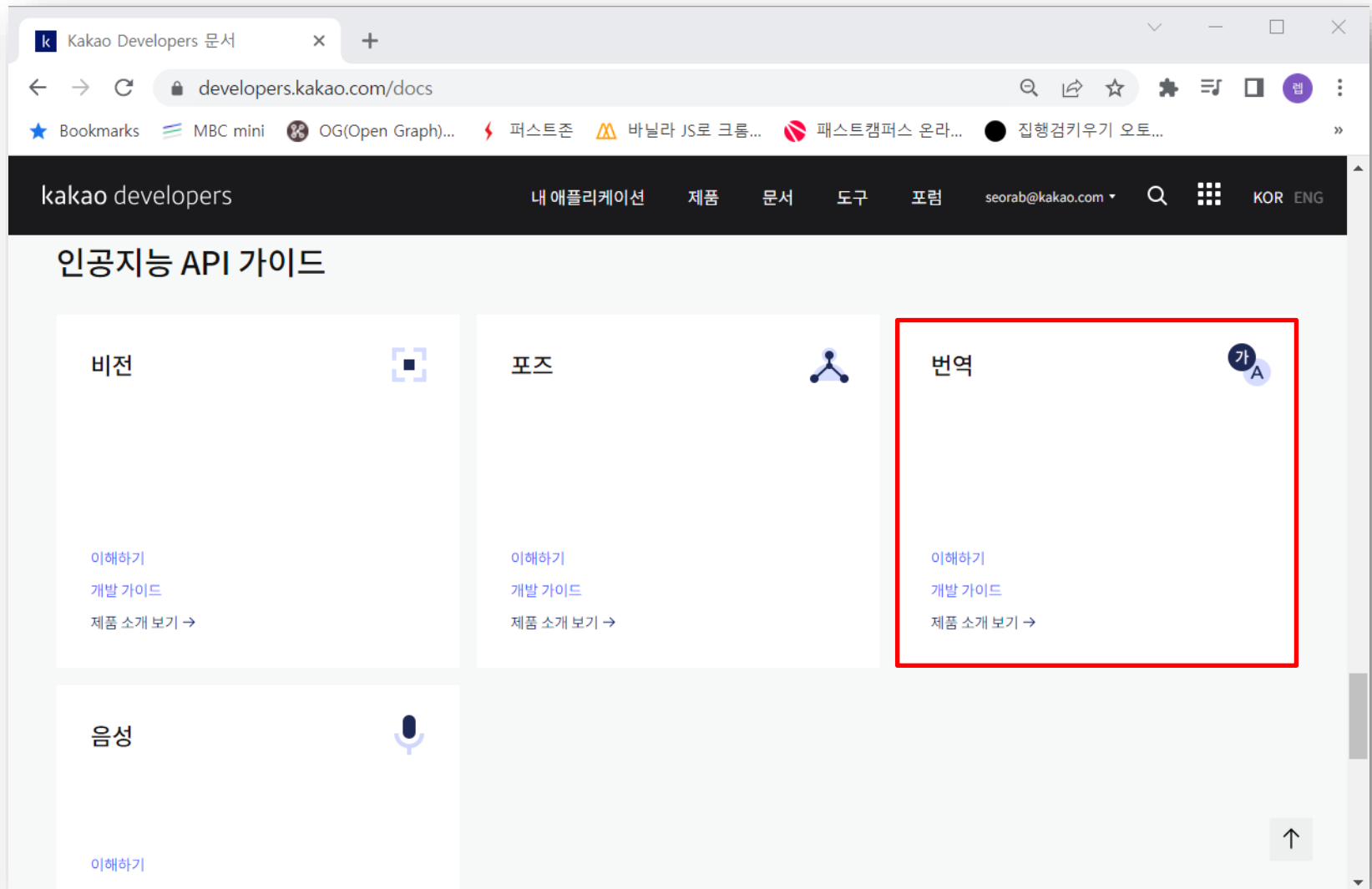
앱 키	
네이티브 앱 키	d5d3d3626499a6c2e4915de477faefe3
REST API 키	28ab70ec0b9cf7dfa1d84a5ce6dade62
JavaScript 키	46f758933e4b8abd7b97abaa9285e6af
Admin 키	ba08fb9bb0d967e69b14aedbd4a8e2a4

Below the table, the "플랫폼" (Platform) section is visible.

## ■ Rest API 활용 프로그램 만들기

### 3. 카카오 개발자 사이트 가입 / 로그인 / API 문서

문서





## ■ Rest API 활용 프로그램 만들기

### 3. 카카오 개발자 사이트 가입 / 로그인 / API 문서

문서

The screenshot shows a web browser window with the address bar displaying `developers.kakao.com/docs/latest/ko/translate/dev-guide`. The page title is "문장 번역하기" (Sentence Translation). The main content area explains the translation API and provides the request details.

문장 번역하기

문장 번역요청한 문장을 다양한 언어로 번역하는 API로써 입력된 텍스트를 기반으로 번역 텍스트 결과를 전달합니다. 한국어와 타언어간 번역외에도 타언어간 번역도 지원됩니다.

REST API 키를 `Authorization` 헤더에 담아 `GET` 또는 `POST` 로 요청합니다. 요청이 성공하면 `JSON` 객체로 번역 결과 목록을 전달합니다.

Request

URL

```
GET /v2/translation/translate HTTP/1.1
Host: dapi.kakao.com
Authorization: KakaoAK ${REST_API_KEY}
```

```
POST /v2/translation/translate HTTP/1.1
Host: dapi.kakao.com
Authorization: KakaoAK ${REST_API_KEY}
Content-type: application/x-www-form-urlencoded
```

## ■ Rest API 활용 프로그램 만들기

### 3. 카카오 개발자 사이트 가입 / 로그인 / API 문서

문서

The screenshot shows a web browser window with the URL `developers.kakao.com/docs/latest/ko/translate/dev-guide`. The page is titled "kakao developers" and has a navigation bar with links for "내 애플리케이션", "제품", "문서", "도구", and "포럼". The main content area is titled "문서 > 번역 > 개발 가이드" and contains a "Parameter" table, a "Response" table, and a "Sample" section.

**Parameter**

Name	Type	Description	Required
query	String	번역 대상 문장 최대 5,000자	O
src_lang	String	번역 대상 언어	O
target_lang	String	번역 결과 언어	O

**Response**

Name	Type	Description
translated_text	List of String[]	번역결과 리스트, 문단 단위의 리스트가 문단 순서대로 정렬

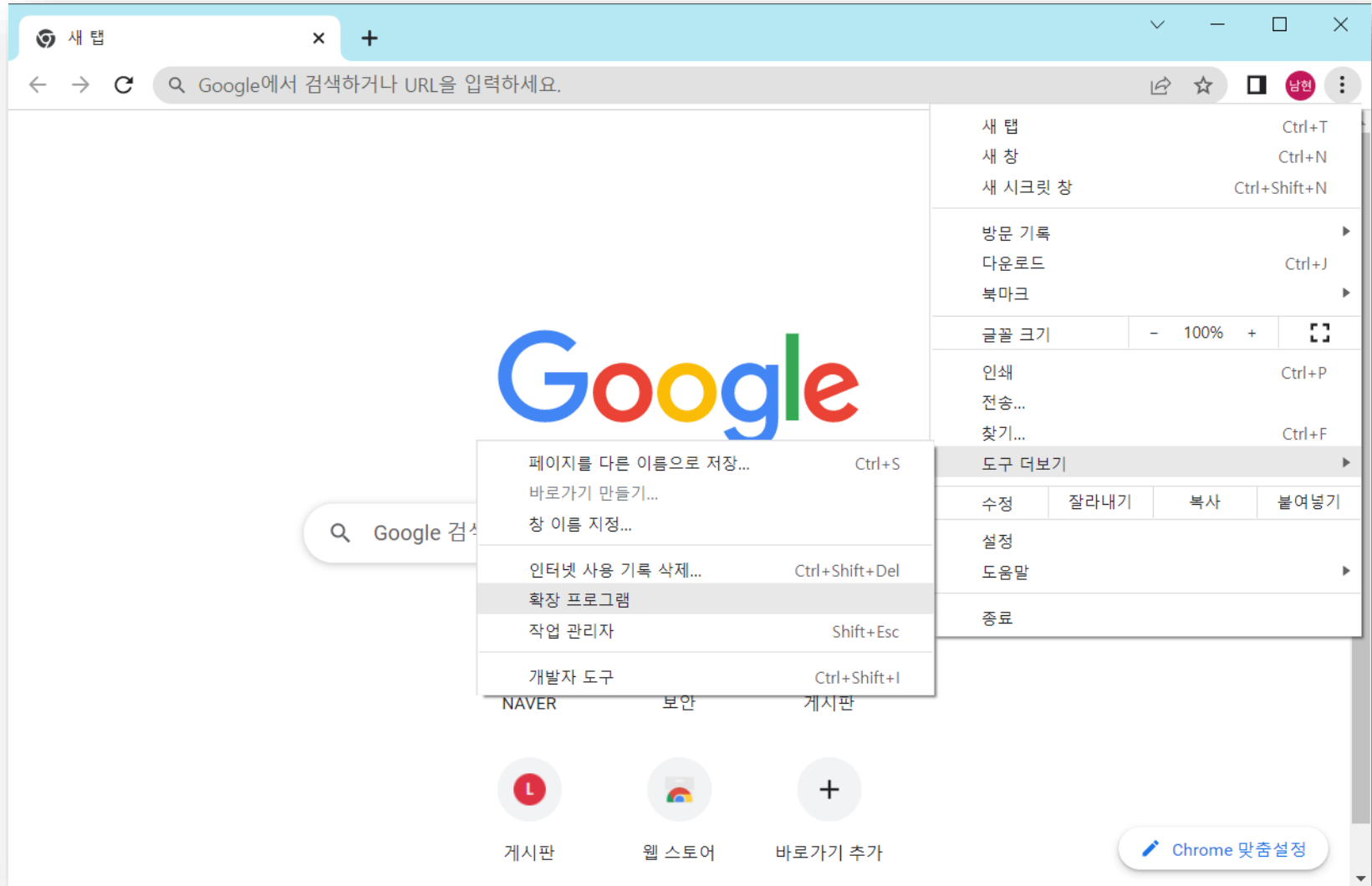
**Sample**

Request: GET 방식

## ■ Rest API 활용 프로그램 만들기

### 4. 웹 브라우저 확장 프로그램 설치 (Json Formatter / API Tester)

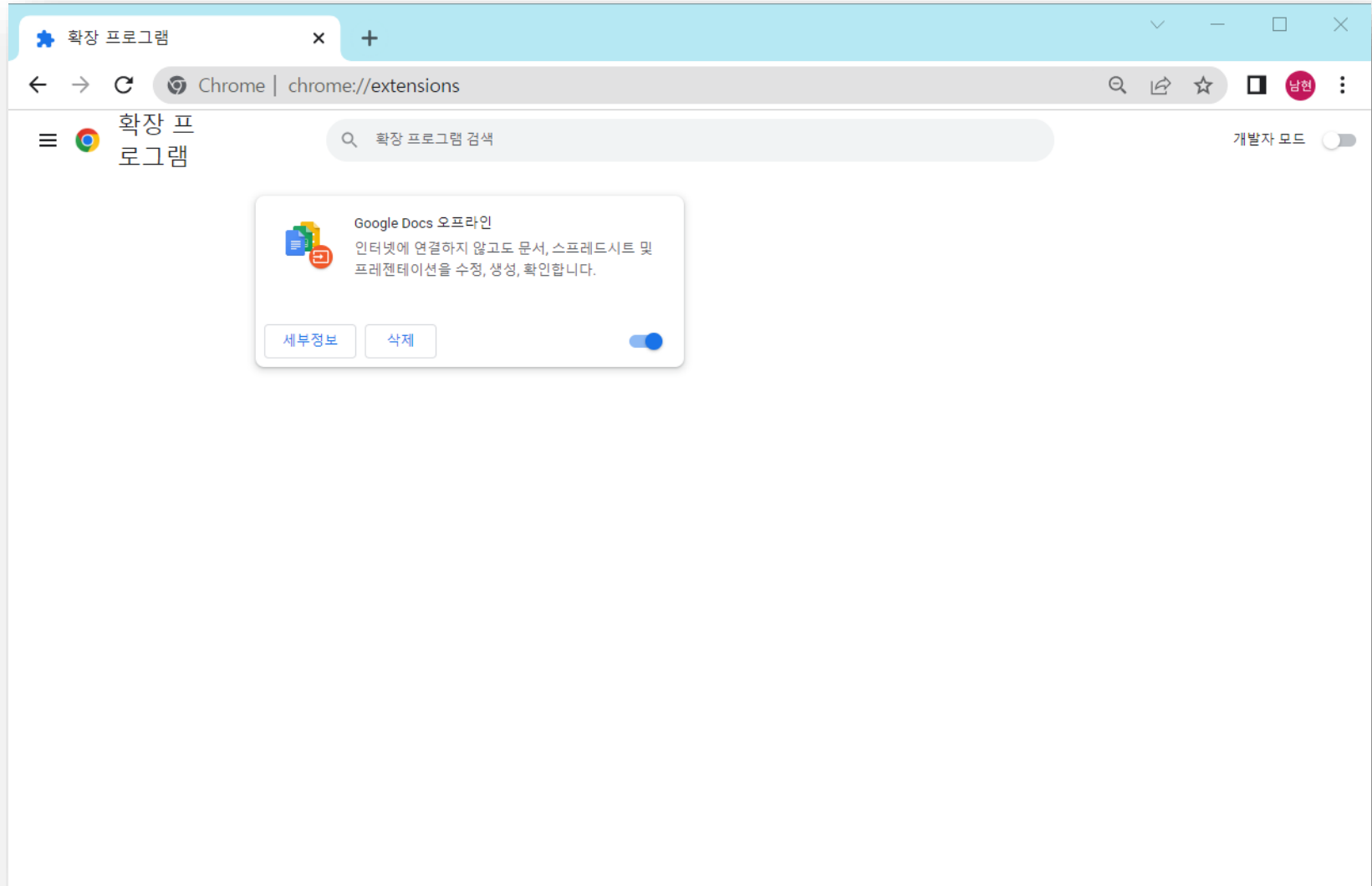
도구 더보기 → 확장 프로그램



## ■ Rest API 활용 프로그램 만들기

### 4. 웹 브라우저 확장 프로그램 설치 (Json Formatter / API Tester)

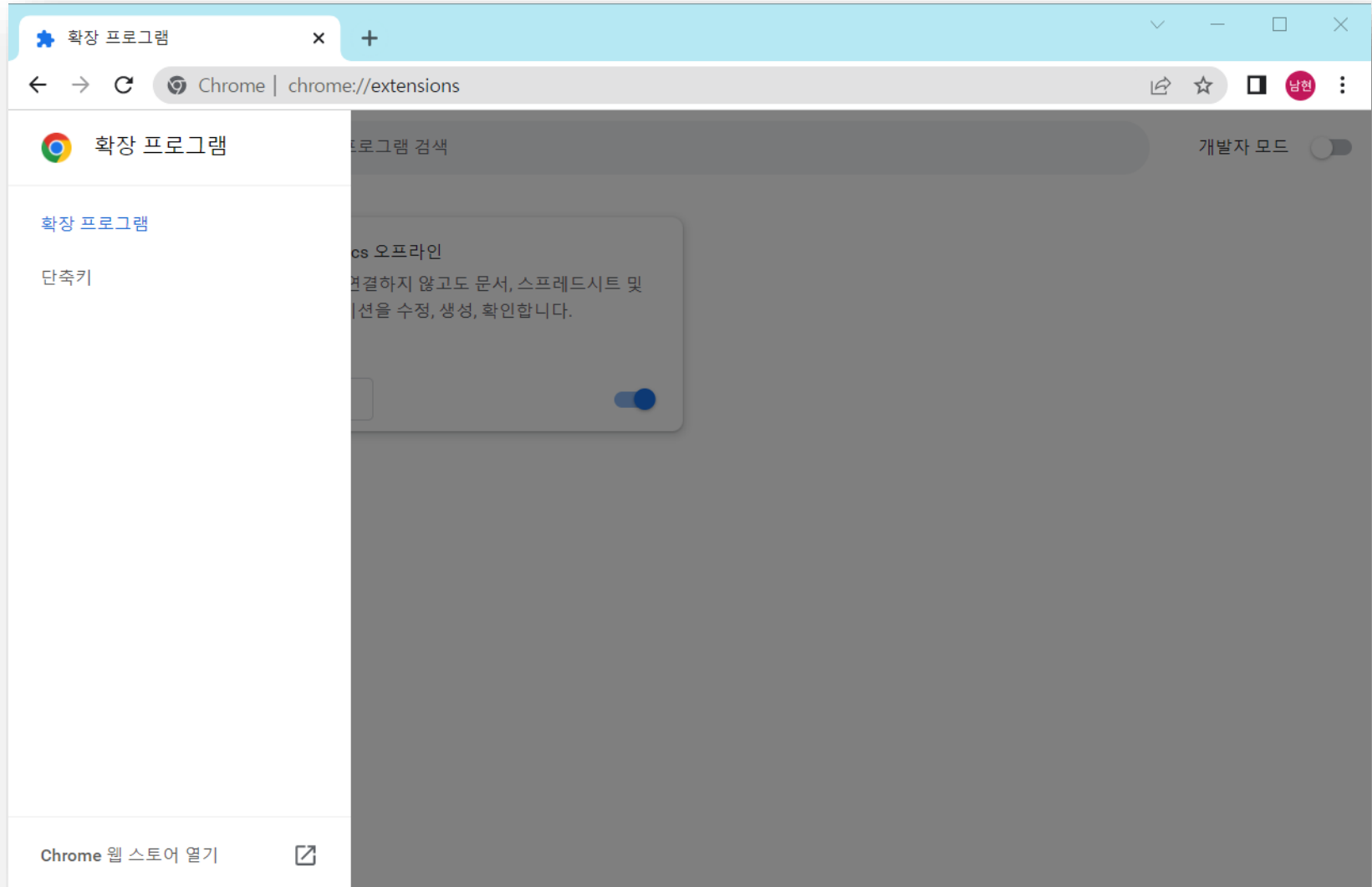
도구 더보기 → 확장 프로그램



## ■ Rest API 활용 프로그램 만들기

#### 4. 웹 브라우저 확장 프로그램 설치 (Json Formatter / API Tester)

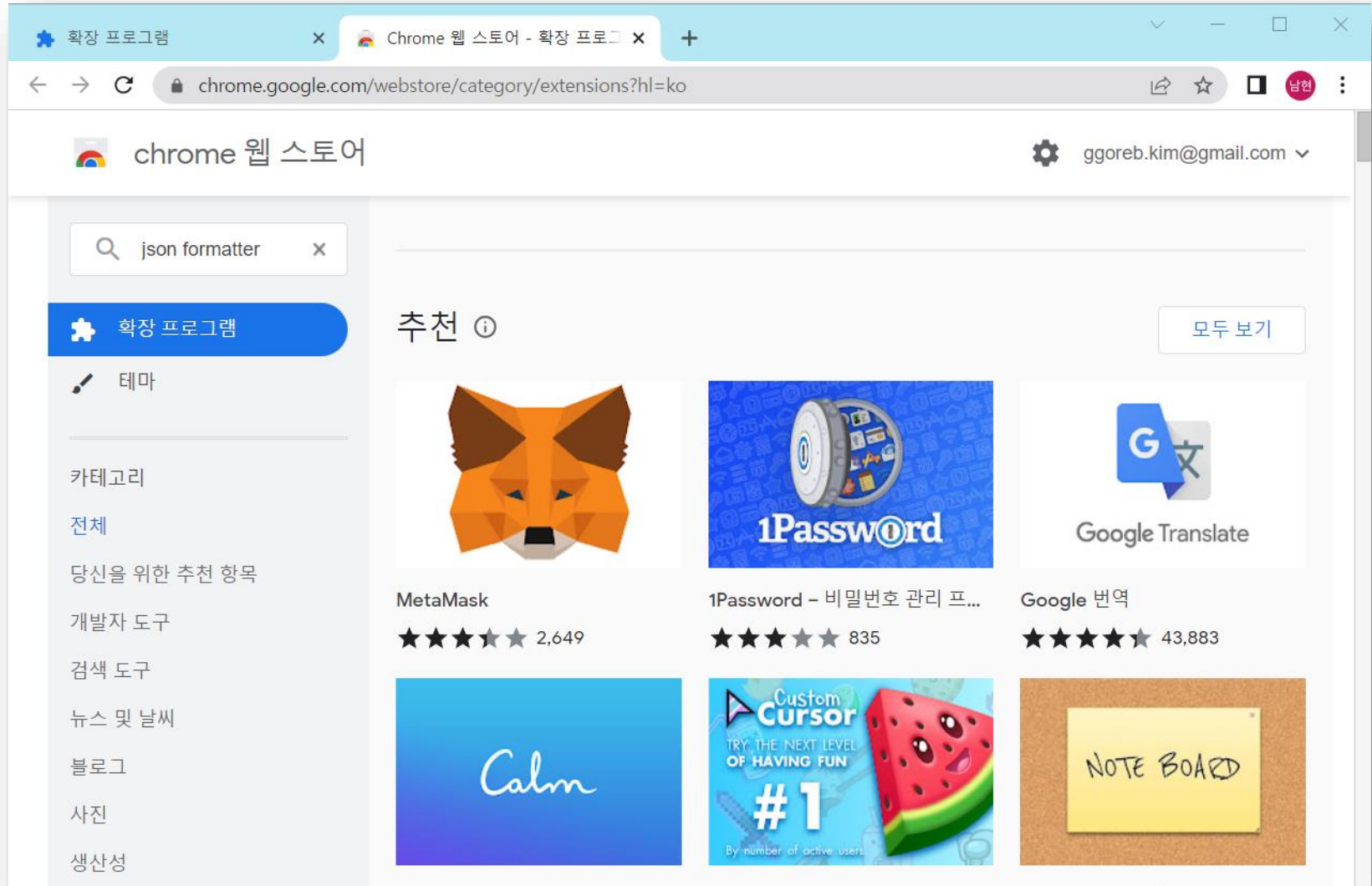
## 도구 더보기 → 확장 프로그램



## ■ Rest API 활용 프로그램 만들기

### 4. 웹 브라우저 확장 프로그램 설치 (Json Formatter / API Tester)

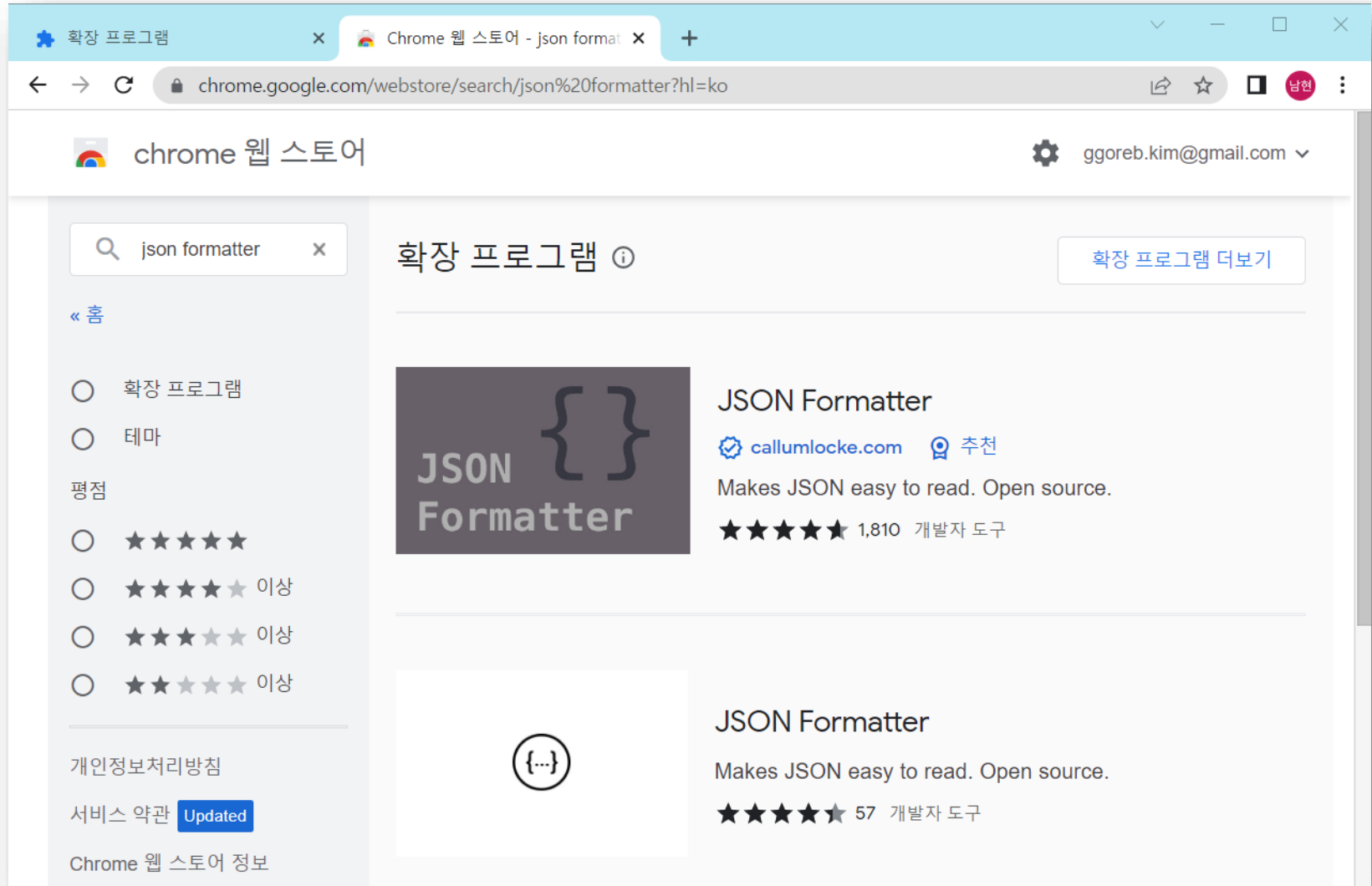
도구 더보기 → 확장 프로그램



## ■ Rest API 활용 프로그램 만들기

### 4. 웹 브라우저 확장 프로그램 설치 (Json Formatter / API Tester)

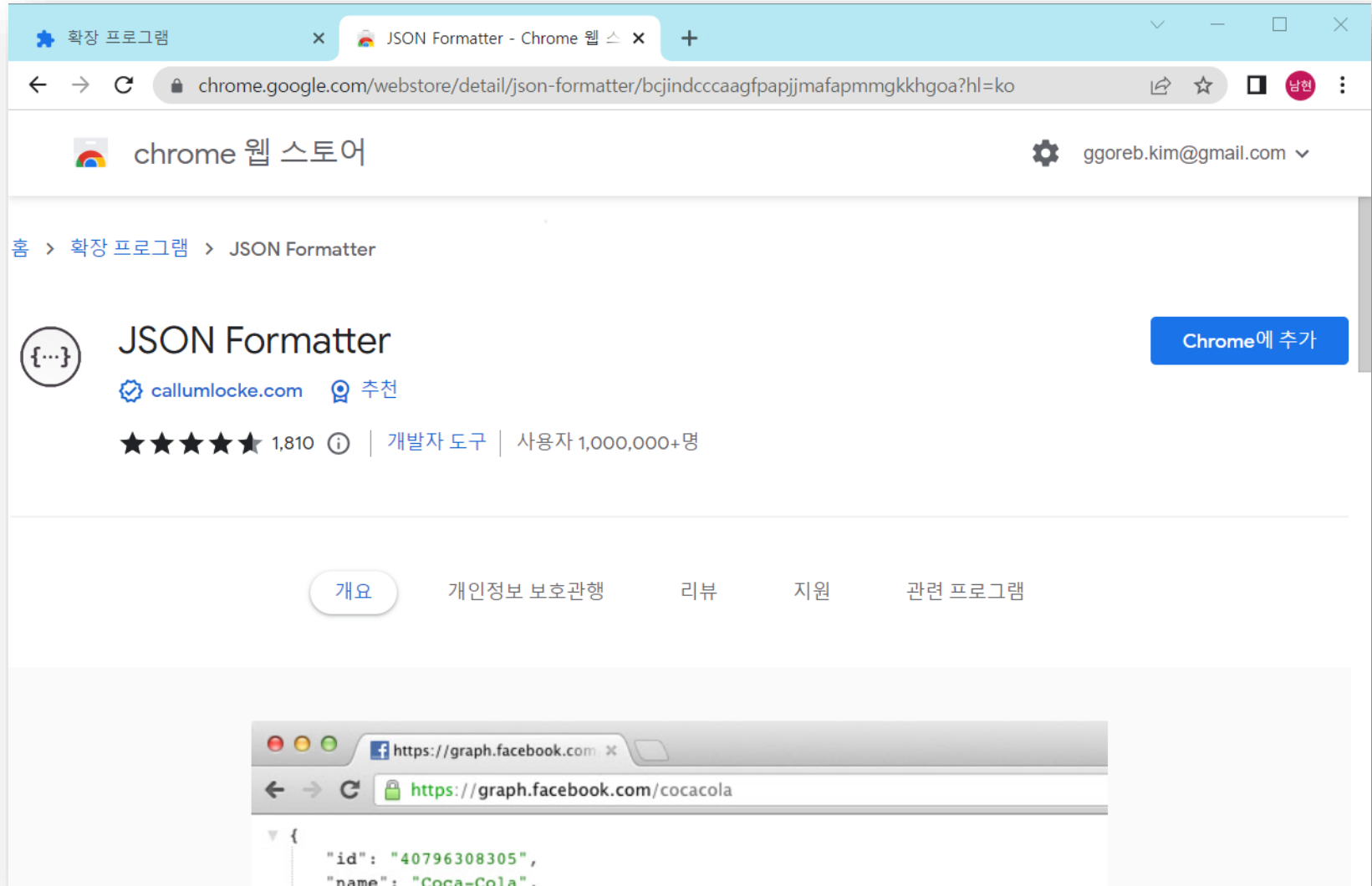
도구 더보기 → 확장 프로그램



## ■ Rest API 활용 프로그램 만들기

### 4. 웹 브라우저 확장 프로그램 설치 (Json Formatter / API Tester)

도구 더보기 → 확장 프로그램





## ■ Rest API 활용 프로그램 만들기

### 4. 웹 브라우저 확장 프로그램 설치 (Json Formatter / API Tester)

도구 더보기 → 확장 프로그램

chrome 웹 스토어

홈 > 확장 프로그램 > JSON Formatter

**JSON Formatter**  
callumlocke.com 추천  
★★★★★ 1,810 | 개발자 도구 | 사용자 1,000,000+명

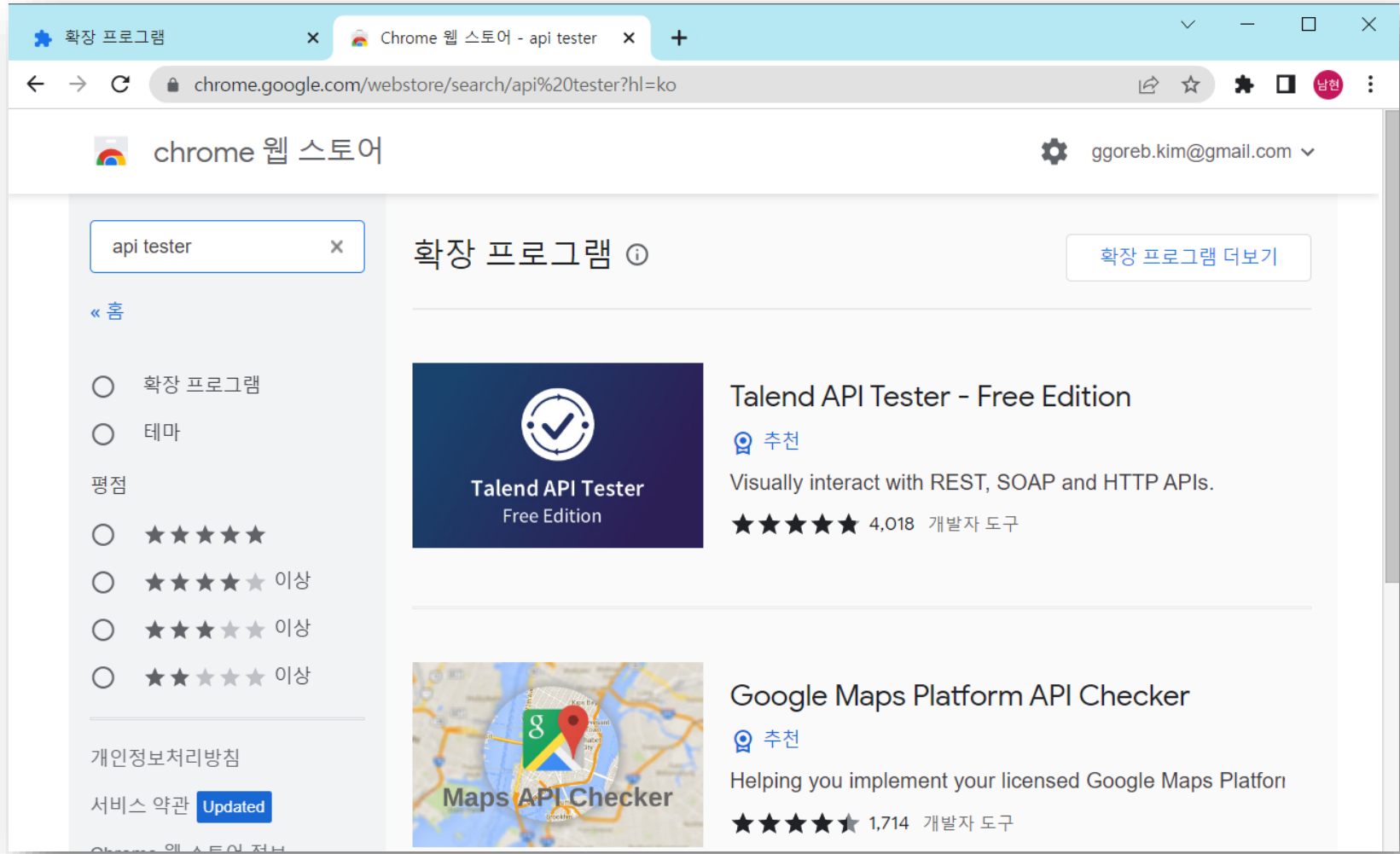
개요 개인정보 보호관행 리뷰 지원 **관련 프로그램**

JSONVue	JSON Beautifier & Editor	JSON Viewer	JSON Beautifier
★★★★★ 2,857	★★★★★ 20	★★★★★ 973	★★★★★ 9

## ■ Rest API 활용 프로그램 만들기

### 4. 웹 브라우저 확장 프로그램 설치 (Json Formatter / API Tester)

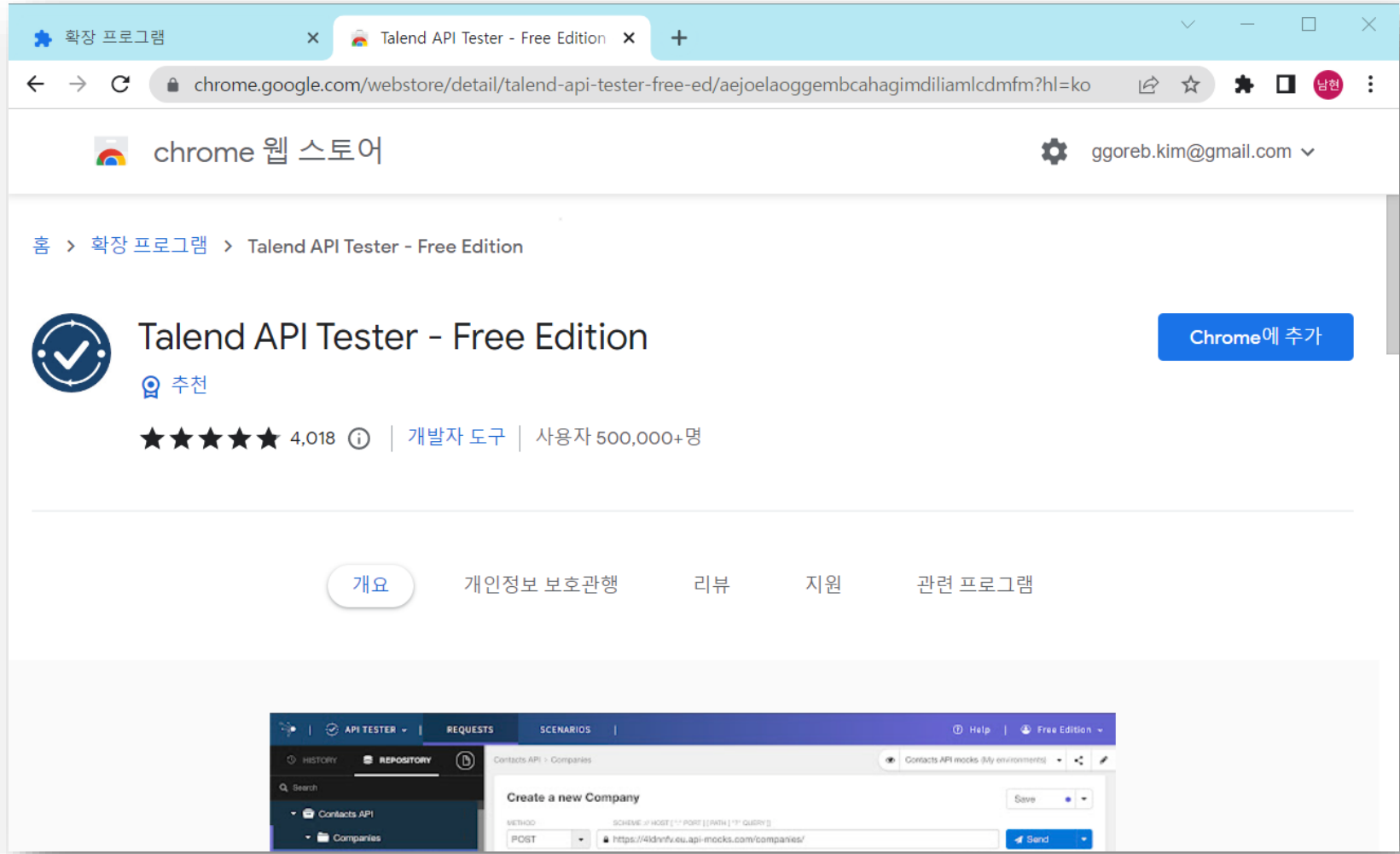
도구 더보기 → 확장 프로그램



## ■ Rest API 활용 프로그램 만들기

### 4. 웹 브라우저 확장 프로그램 설치 (Json Formatter / API Tester)

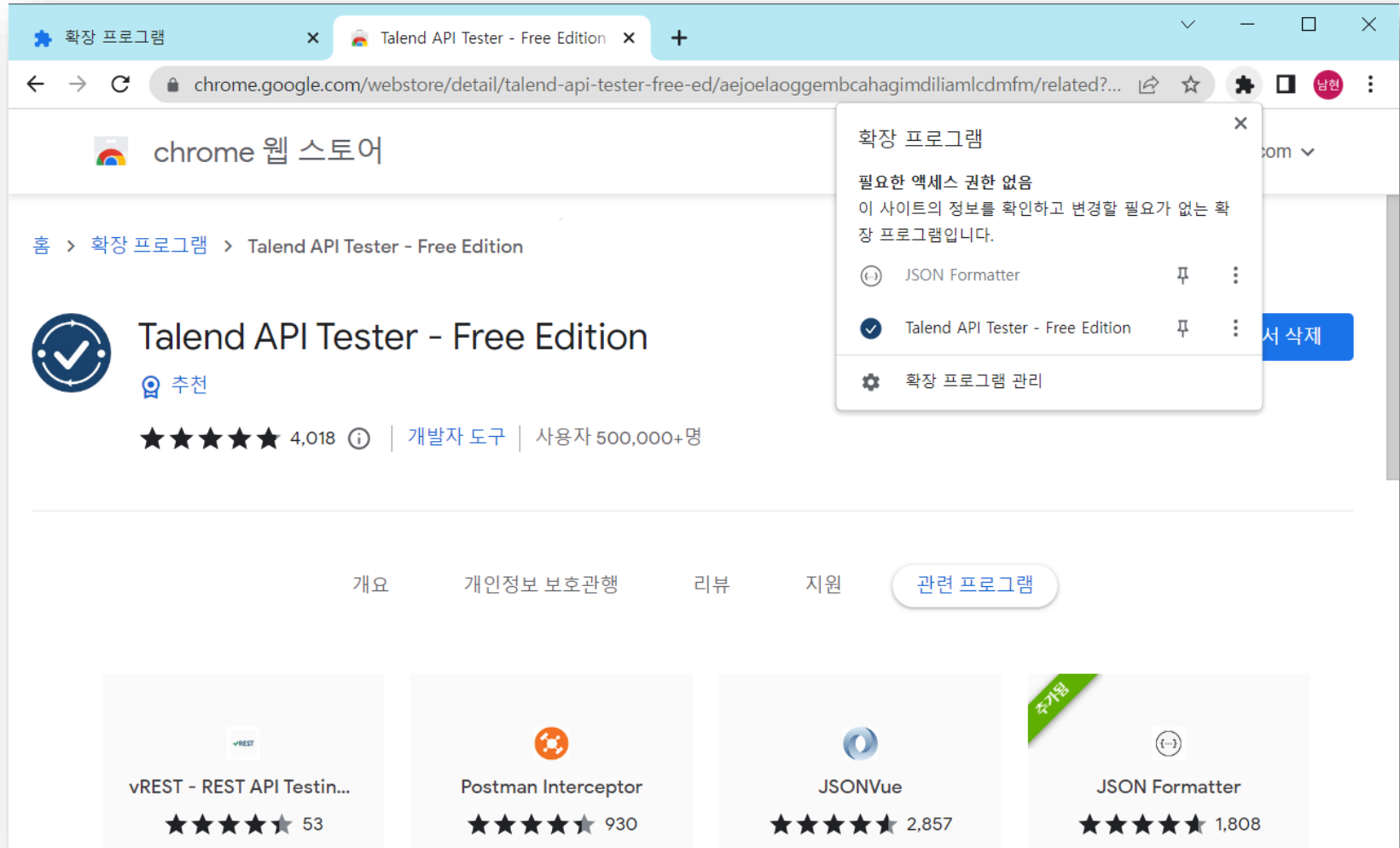
도구 더보기 → 확장 프로그램



## ■ Rest API 활용 프로그램 만들기

### 4. 웹 브라우저 확장 프로그램 설치 (Json Formatter / API Tester)

도구 더보기 → 확장 프로그램



chrome 웹 스토어

홈 > 확장 프로그램 > Talend API Tester - Free Edition

**Talend API Tester - Free Edition**

추천

★★★★★ 4,018 ⓘ | 개발자 도구 | 사용자 500,000+명

개요 개인정보 보호관행 리뷰 지원 **관련 프로그램**

Extension Name	Rating
vREST - REST API Testin...	★★★★★ 53
Postman Interceptor	★★★★★ 930
JSONVue	★★★★★ 2,857
JSON Formatter	★★★★★ 1,808

## ■ Rest API 활용 프로그램 만들기

### 5. ObjectMapper 라이브러리 사용법 - Map

```
ObjectMapper objectMapper = new ObjectMapper();

String json = "{\"id\": 123, \"pw\": 456}";

Map<String, Object> map =
    objectMapper.readValue(json, Map.class);
System.out.println(map);
```

```
{
  "id": 123,
  "pw": 456
}
```

```
{id=123, pw=456}
```

## ■ Rest API 활용 프로그램 만들기

### 5. ObjectMapper 라이브러리 사용법 - VO (Value Object)

```
ObjectMapper objectMapper = new ObjectMapper();

String json = "{\"id\": 123, \"pw\": 456}";

JsonVO map =
    objectMapper.readValue(json, JsonVO.class);
System.out.println(map);
```

```
public class JsonVO {
    int id;
    int pw;

    public int getId() { return id; }
    public void setId(int id) { this.id = id; }

    public int getPw() { return pw; }
    public void setPw(int pw) { this.pw = pw; }

    @Override
    public String toString() {
        return "JsonVO [id=" + id + ", pw=" + pw + "]";
    }
}
```

## ■ Rest API 활용 프로그램 만들기

### 5. ObjectMapper 라이브러리 사용법 - Map

```
ObjectMapper objectMapper = new ObjectMapper();  
  
URL url = new URL("http://gcoreb.com/http/json1.jsp");  
  
Map map = objectMapper.readValue(url, Map.class);  
System.out.println(map);
```

```
▼ {  
    "age": 10,  
    "name": "GGoReb"  
}
```

```
{age=10, name=GGoReb}
```

## ■ Rest API 활용 프로그램 만들기

### 5. ObjectMapper 라이브러리 사용법 - VO

```
ObjectMapper objectMapper = new ObjectMapper();

URL url = new URL("http://ggoreb.com/http/json1.jsp");

JsonVO map = objectMapper.readValue(url, JsonVO.class);
System.out.println(map);
```

```
public class JsonVO {
    String name;
    int age;

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public int getAge() { return age; }
    public void setAge(int age) { this.age = age; }

    @Override
    public String toString() {
        return "JsonVO [name=" + name + ", age=" + age + "]";
    }
}
```



## ■ Rest API 활용 프로그램 만들기

### 5. ObjectMapper 라이브러리 사용법 - List

```
ObjectMapper objectMapper = new ObjectMapper();  
  
URL url = new URL("http://ggoreb.com/http/json2.jsp");  
  
List list = objectMapper.readValue(url, List.class);  
System.out.println(list);
```

```
[{age=10, name=A}, {age=11, name=B}, {age=12, name=C}]
```

```
▼ [  
  ▼ {  
    "age": 10,  
    "name": "A"  
  },  
  ▼ {  
    "age": 11,  
    "name": "B"  
  },  
  ▼ {  
    "age": 12,  
    "name": "C"  
  }  
]
```

## ■ Rest API 활용 프로그램 만들기

### 5. ObjectMapper 라이브러리 사용법 - VO

```
ObjectMapper objectMapper = new ObjectMapper();

URL url = new URL("http://ggoreb.com/http/json2.jsp");

JsonVO[] list = objectMapper.readValue(url, JsonVO[].class);

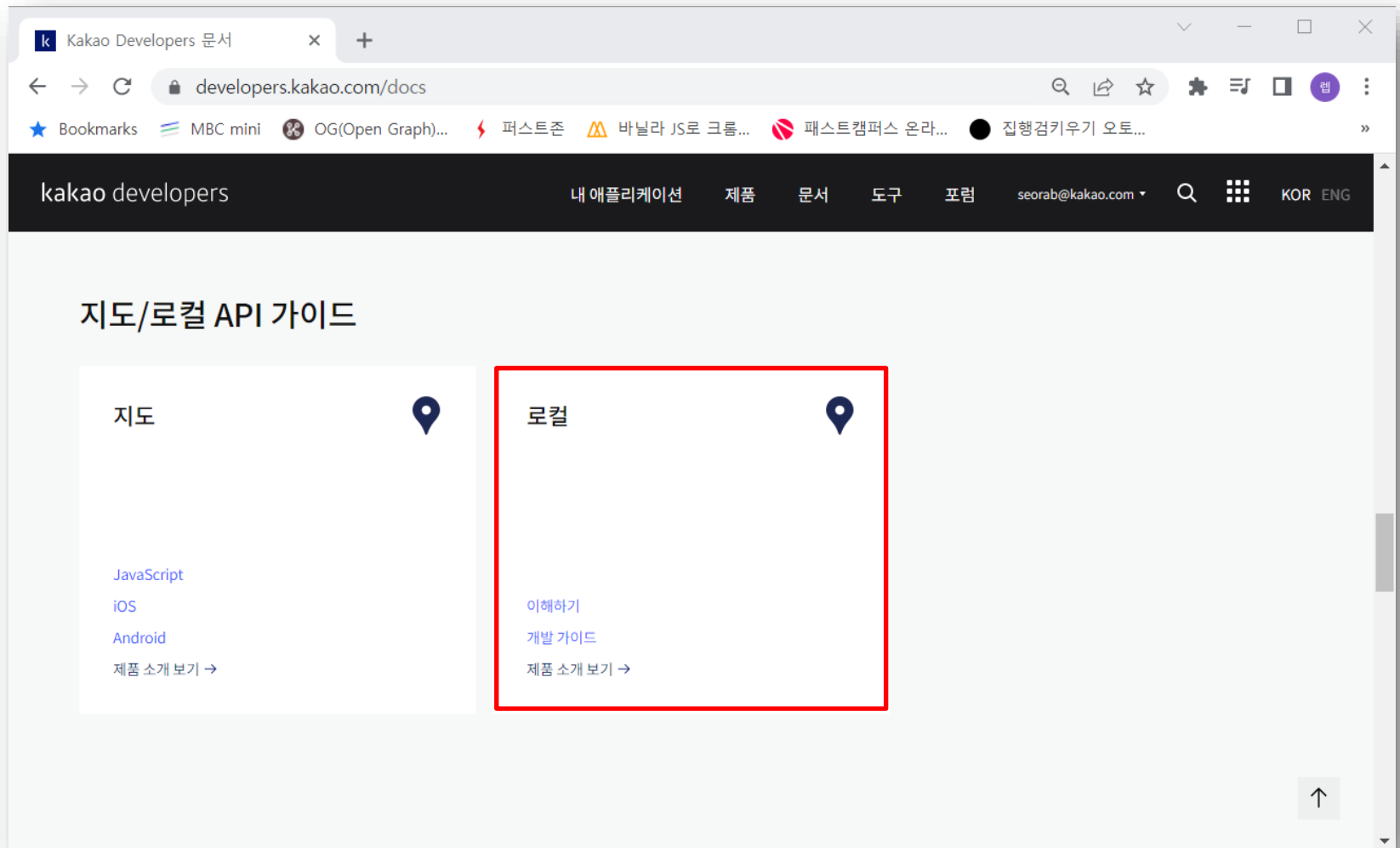
for(JsonVO vo : list) {
    System.out.println(vo);
}
```

```
JsonVO [name=A, age=10]
JsonVO [name=B, age=11]
JsonVO [name=C, age=12]
```

## ■ Rest API 활용 프로그램 만들기

### ● ex.AddressKakaoAPI

문서



# Rest API 활용 프로그램 만들기

## ex.AddressKakaoAPI

문서

kakao developers

문서 > 로컬 > 개발 가이드

Request

URL

```
GET /v2/local/search/address.${FORMAT} HTTP/1.1
Host: dapi.kakao.com
Authorization: KakaoAK ${REST_API_KEY}
```

Parameter

Name	Type	Description	Required
query	String	검색을 원하는 질의어	O
analyze_type	String	검색 결과 제공 방식 다음 중 하나: similar: 입력한 건물명과 일부만 매칭될 경우에도 확장된 검색 결과 제공, 미지정 시 similar가 적용됨 exact: 주소의 정확한 건물명이 입력된 주소패턴일 경우에 한해, 입력한 건물명과 정확히 일치하는 검색 결과 제공 (기본값: similar)	X

# Rest API 활용 프로그램 만들기

## ex.AddressKakaoAPI

문서

The screenshot shows the Kakao Developers API documentation page. The left sidebar contains a list of API categories. The main content area displays the 'Response' section, which is highlighted with a red box. Below the 'Response' header, there are two sections: 'meta' and 'documents'. Each section contains a table with columns for Name, Type, and Description.

**Response**

**meta**

Name	Type	Description
total_count	Integer	검색어에 검색된 문서 수
pageable_count	Integer	total_count 중 노출 가능 문서 수 (최대: 45)
is_end	Boolean	현재 페이지가 마지막 페이지인지 여부 값이 false면 다음 요청 시 page 값을 증가시켜 다음 페이지 요청 가능

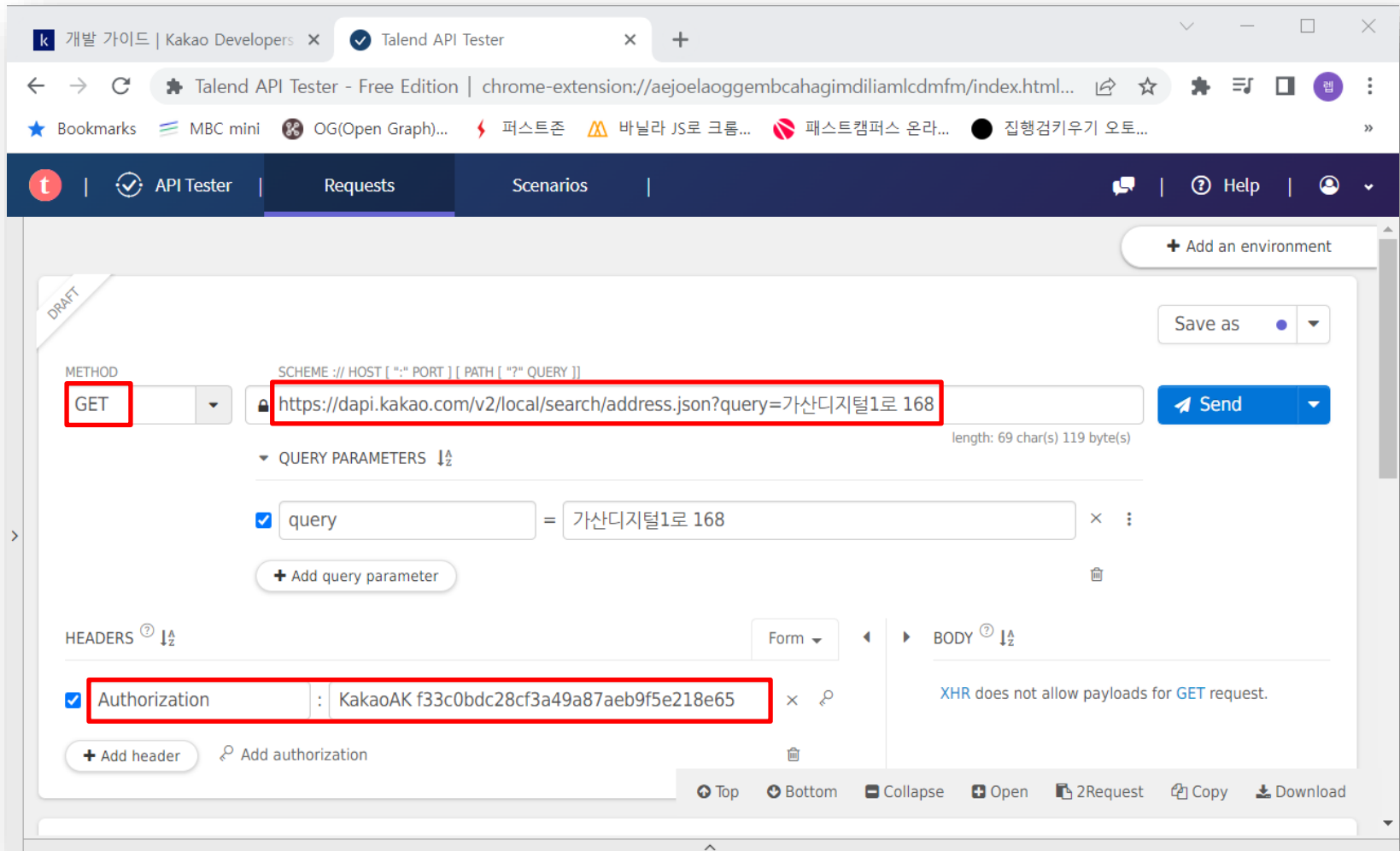
**documents**

Name	Type	Description
address_name	String	전체 지번 주소 또는 전체 도로명 주소, 입력에 따라 결정됨

## ■ Rest API 활용 프로그램 만들기

### ● ex.AddressKakaoAPI

#### 확장프로그램 API Tester



## ■ Rest API 활용 프로그램 만들기

### ● ex.AddressKakaoAPI

#### 확장프로그램 API Tester

The screenshot displays the Talend API Tester interface within a web browser. The browser's address bar shows the URL: `chrome-extension://aejoelaoggembcahagimdiliamlcdmfm/index.html...`. The interface has a dark blue header with the 'API Tester' tab selected. Below the header, a green banner indicates a '200 OK' status. The main area is split into two panels: 'HEADERS' on the left and 'BODY' on the right. The 'HEADERS' panel lists various response headers such as 'Server: nginx', 'Date: Sat, 18 Jun 2022 22:38:04 GMT +1s', and 'Content-Type: application/json; charset=UTF-8'. The 'BODY' panel shows a JSON response, which is highlighted with a red rectangle. The JSON structure includes a 'documents' array with one object containing 'address' details (name, type, road address) and coordinates (x, y), and a 'meta' object with 'is\_end: true'.

```
1 {
2   documents : [
3     {
4       address : { address_name : "서울 금천구 가산동 371
18       address_name : "서울 금천구 가산디지털1로 168",
19       address_type : "ROAD_ADDR",
20       road_address : { address_name : "서울 금천구 가산C
34       x : "126.88263620024",
35       y : "37.479966429878"
36     }
37   ],
38   meta : {
39     is_end : true,
```

## ■ Rest API 활용 프로그램 만들기

### ● ex.AddressKakaoAPI

```
public static final String restAPIKey = "f33c0bdc28cf3a49a87aeb9f5e218e65";

public static void main(String[] args) throws IOException {
    String query = "부산 연제구 중앙대로 1001";

    String apiURL = "https://dapi.kakao.com/v2/local/search/address.json";
    apiURL += "?query=" + URLEncoder.encode(query, "utf-8");

    URL url = new URL(apiURL);
    HttpURLConnection con = (HttpURLConnection) url.openConnection();
    con.setRequestProperty("Authorization", "KakaoAK " + restAPIKey);

    ObjectMapper objectMapper = new ObjectMapper();
    Map<String, Object> object = objectMapper.readValue(con.getInputStream(), Map.class);
    System.out.println(object);

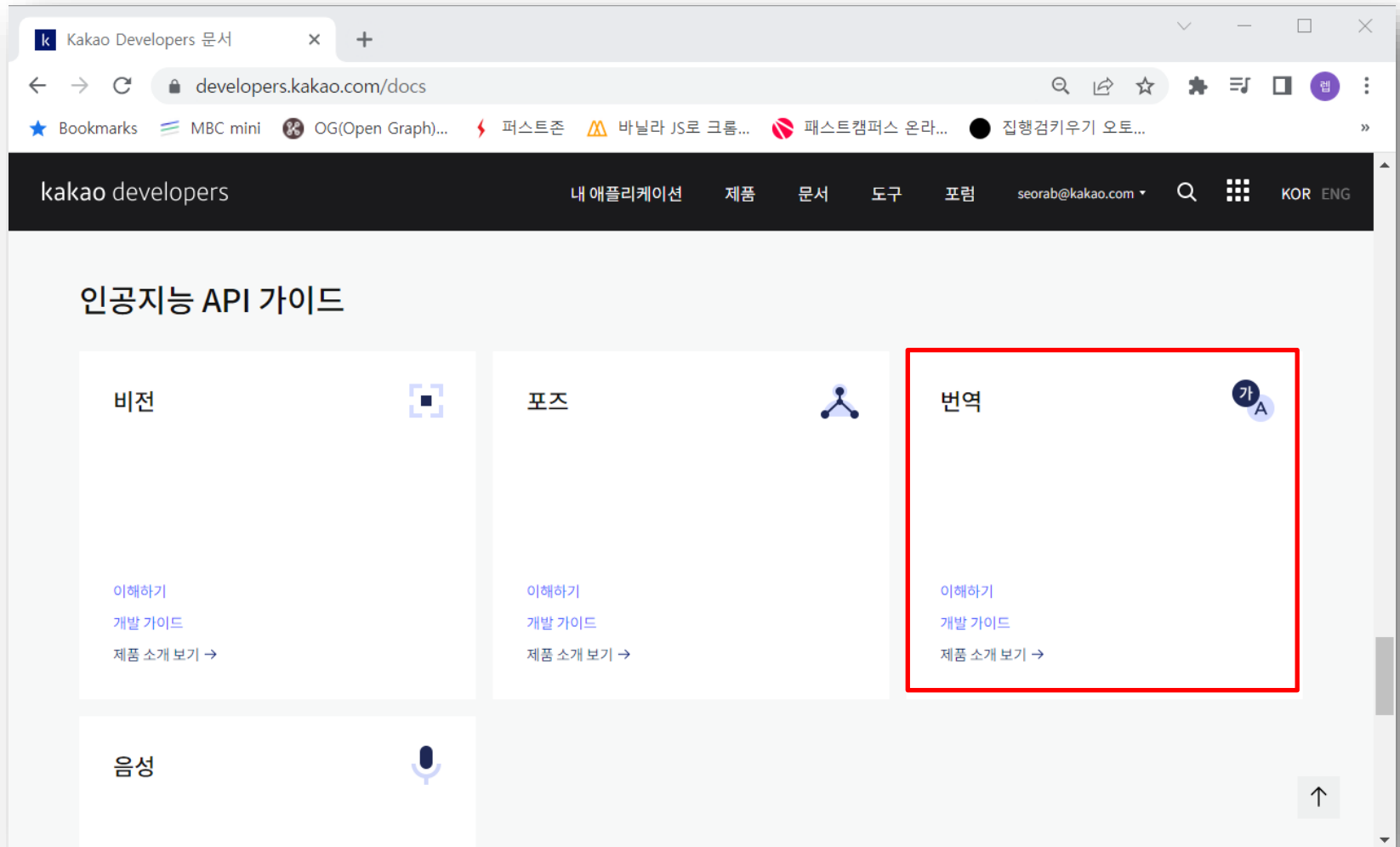
    List<Map> documents = (List<Map>) object.get("documents");
    for (Map document : documents) {
        String addressName = (String) document.get("address_name");
        String latitude = (String) document.get("y");
        String longitude = (String) document.get("x");
        System.out.println(addressName);
        System.out.printf("%s / %s\n", latitude, longitude);
    }
}
```



## ■ Rest API 활용 프로그램 만들기

### ● ex.TranslationKakaoAPI

문서



# ■ Rest API 활용 프로그램 만들기

## ● ex.TranslationKakaoAPI

문서

kakao developers

문서 > 번역 > 개발 가이드

시작하기  
카카오 로그인  
카카오싱크  
카카오톡 소셜  
메시지  
카카오스토리  
카카오톡 채널  
카카오페이  
카카오모먼트  
카카오 키워드광고  
푸시 알림  
Daum 검색  
지도

**Request**

URL

```
GET /v2/translation/translate HTTP/1.1
Host: dapi.kakao.com
Authorization: KakaoAK ${REST_API_KEY}
```

```
POST /v2/translation/translate HTTP/1.1
Host: dapi.kakao.com
Authorization: KakaoAK ${REST_API_KEY}
Content-type: application/x-www-form-urlencoded
```

Parameter

Name	Type	Description	Required
query	String	번역 대상 문자 최대 5,000자	0

# ■ Rest API 활용 프로그램 만들기

## ● ex.TranslationKakaoAPI

문서

kakao developers

문서 > 번역 > 개발 가이드

시작하기  
카카오 로그인  
카카오싱크  
카카오톡 소셜  
메시지  
카카오스토리  
카카오톡 채널  
카카오페이  
카카오모먼트  
카카오 키워드광고  
푸시 알림  
Daum 검색  
지도

**Response**

Name	Type	Description
translated_text	List of String[]	번역결과 리스트, 문단 단위의 리스트가 문단 순서대로 정렬

**Sample**

Request: GET 방식

```
curl -v --get "https://dapi.kakao.com/v2/translation/translate" \
-H "Authorization: KakaoAK ${REST_API_KEY}" \
-d "src_lang=kr" \
-d "target_lang=en" \
--data-urlencode "query=지난해 3월 오픈한 카카오톡 주문하기는 현재까지 약 250만명의 회원들"
```

Request: POST 방식

## ■ Rest API 활용 프로그램 만들기

### ● ex.TranslationKakaoAPI

#### 확장프로그램 API Tester

The screenshot shows the Talend API Tester interface with the following configuration:

- METHOD:** GET
- URL:** `https://dapi.kakao.com/v2/translation/translate?src_lang=kr&target_lang=en&query=카카오 개발자 사(`
- QUERY PARAMETERS:**
  - `src_lang` = `kr`
  - `target_lang` = `en`
  - `query` = `카카오 개발자 사이트에서 제공되는 기능 활용하기`
- HEADERS:**
  - `Authorization` : `KakaoAK f33c0bdc28cf3a49a87aeb9f5e218e65`

The interface also includes a "Send" button, a "Save as" dropdown, and a "BODY" section with a message: "XHR does not allow payloads for GET request."

# ■ Rest API 활용 프로그램 만들기

## ● ex.TranslationKakaoAPI

### 확장프로그램 API Tester

The screenshot displays the Talend API Tester interface within a web browser. The browser's address bar shows the URL: `chrome-extension://aejoelaoggembcahagimdiliamlcdmfm/index.html...`. The interface has a dark blue header with tabs for 'API Tester', 'Requests', and 'Scenarios'. The 'API Tester' tab is active, showing a green status bar at the top with '200 OK'. Below this, the response is divided into two panels: 'HEADERS' on the left and 'BODY' on the right. The 'HEADERS' panel lists various response headers such as 'Server: nginx', 'Date: Sat, 18 Jun 2022 22:57:42 GMT', 'Content-Type: application/json', and 'Access-Control-Allow-Origin: \*'. The 'BODY' panel shows a JSON response with a 'translated\_text' array containing a single string: 'Utilize the features provided on the Kakao Developer...'. A red rectangle highlights the JSON body content. At the bottom of the interface, there are buttons for 'Top', 'Bottom', 'Collapse', 'Open', '2Request', 'Copy', and 'Download'.

200 OK

HEADERS ②

pretty

Server: nginx  
Date: Sat, 18 Jun 2022 22:57:42 GMT  
Content-Type: application/json  
Content-Length: 83 bytes  
Connection: keep-alive  
X-Request-Id: 0f6e8880-ef5a-11ec-9d20-819f05feaf5a  
x-metering-cou... 26  
Access-Control... \*  
Access-Control... GET, POST, PUT, DELETE, OPTIONS  
Access-Control... Authorization, KA, Origin, X-Request  
ed-With, Content-Type, Accept

▶ COMPLETE REQUEST HEADERS

BODY ②

pretty

```
1 {  
2   translated_text : [  
3     [  
4       "Utilize the features provided on the Kakao Developer :  
5     ]  
6   ]  
7 }
```

hide copy length: 83 bytes

Top Bottom Collapse Open 2Request Copy Download

## ■ Rest API 활용 프로그램 만들기

### ● ex.TranslationKakaoAPI - GET

```
String source = "kr"; // 번역 대상 언어
String target = "en"; // 번역 결과 언어
String query = "94년 제가 LA에 처음 갔을 때 모든 경기..."; // 번역 대상 문장

String apiURL = "https://dapi.kakao.com/v2/translation/translate";
apiURL += "?src_lang=" + source;
apiURL += "&target_lang=" + target;
apiURL += "&query=" + URLEncoder.encode(query, "utf-8");

URL url = new URL(apiURL);
URLConnection con = (URLConnection) url.openConnection();
con.setRequestProperty("Authorization", "KakaoAK " + restAPIKey);

ObjectMapper objectMapper = new ObjectMapper();
Map<String, Object> object = objectMapper.readValue(con.getInputStream(), Map.class);
System.out.println(object);

List<List> list = (List<List>) object.get("translated_text");
for(List<String> li : list) {
    for(String text : li) {
        System.out.println(text);
    }
}
```

## ■ Rest API 활용 프로그램 만들기

### ● ex.TranslationKakaoAPI - POST

```
String apiURL = "https://dapi.kakao.com/v2/translation/translate";
URL url = new URL(apiURL);
URLConnection con = (URLConnection) url.openConnection();
con.setRequestProperty("Authorization", "KakaoAK " + restAPIKey);

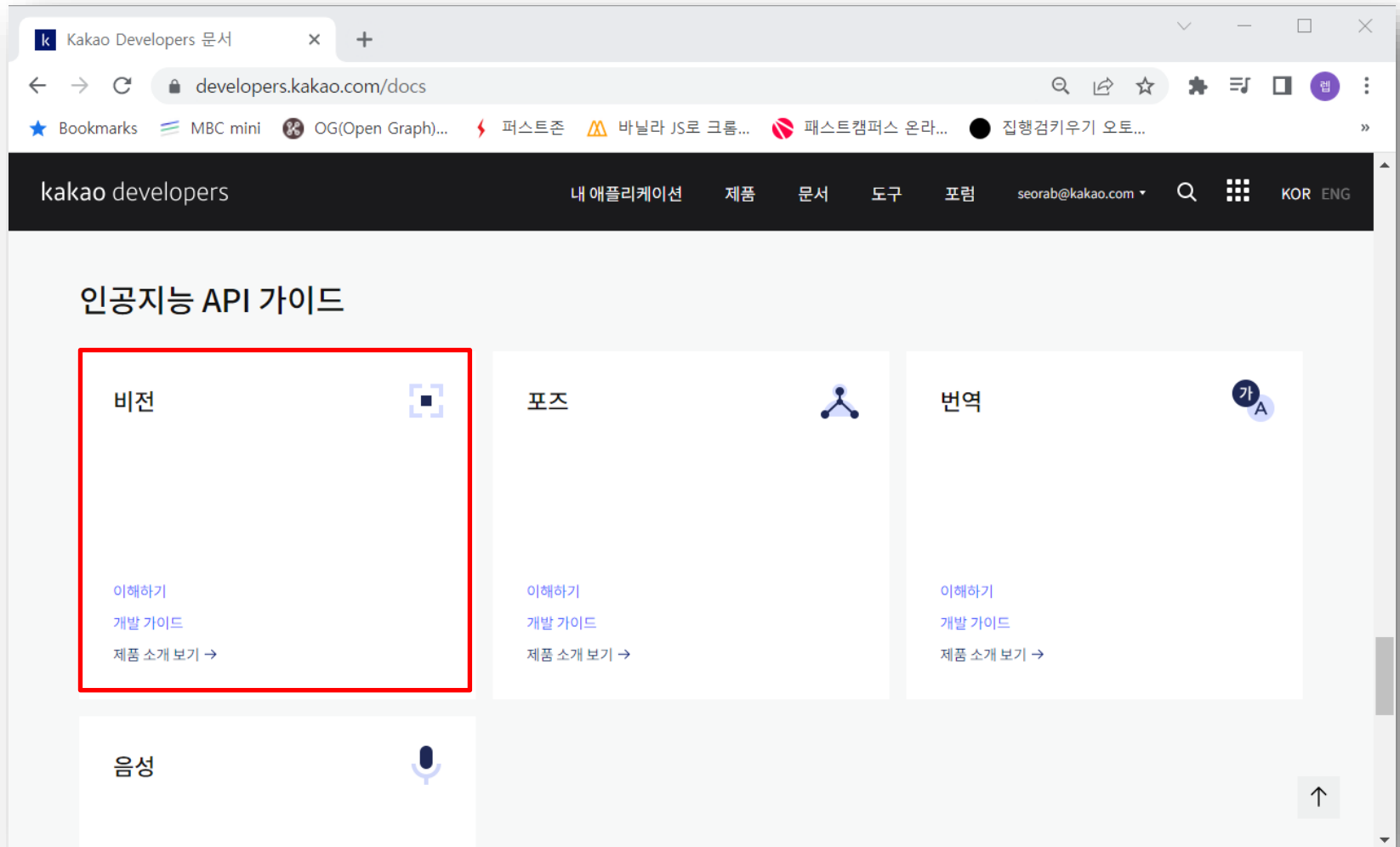
con.setRequestMethod("POST");
con.setDoOutput(true);
con.setDoInput(true);

OutputStream outputStream = con.getOutputStream();
PrintWriter writer =
    new PrintWriter(new OutputStreamWriter(outputStream, "UTF-8"), true);
writer.append("src_lang=" + source);
writer.append("&target_lang=" + target);
writer.append("&query=" + query);
writer.flush();
```

## ■ Rest API 활용 프로그램 만들기

### ● ex.VisionKakaoAPI

문서

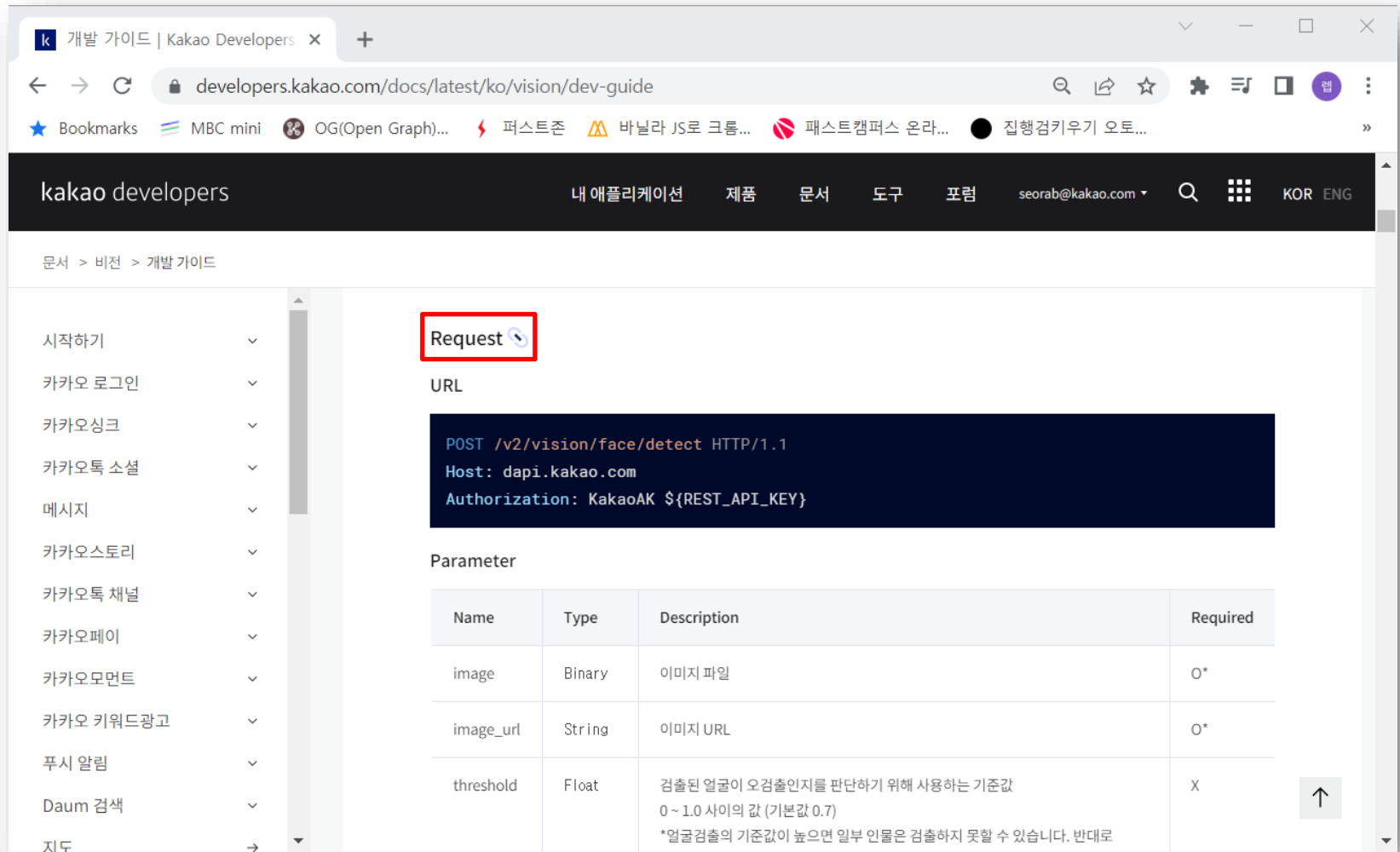




# ■ Rest API 활용 프로그램 만들기

## ● ex.VisionKakaoAPI

문서



kakao developers

문서 > 비전 > 개발 가이드

시작하기  
카카오 로그인  
카카오싱크  
카카오톡 소셜  
메시지  
카카오스토리  
카카오톡 채널  
카카오페이  
카카오모먼트  
카카오 키워드광고  
푸시 알림  
Daum 검색  
지도

Request

URL

```
POST /v2/vision/face/detect HTTP/1.1
Host: dapi.kakao.com
Authorization: KakaoAK ${REST_API_KEY}
```

Parameter

Name	Type	Description	Required
image	Binary	이미지 파일	O*
image_url	String	이미지 URL	O*
threshold	Float	검출된 얼굴이 오검출인지를 판단하기 위해 사용하는 기준값 0 ~ 1.0 사이의 값 (기본값 0.7) *얼굴검출의 기준값이 높으면 일부 인물은 검출하지 못할 수 있습니다. 반대로	X

# Rest API 활용 프로그램 만들기

## ex.VisionKakaoAPI

문서

kakao developers

내 애플리케이션 제품 문서 도구 포럼 seorab@kakao.com KOR ENG

문서 > 비전 > 개발 가이드

시작하기  
카카오 로그인  
카카오싱크  
카카오톡 소셜  
메시지  
카카오스토리  
카카오톡 채널  
카카오페이  
카카오모먼트  
카카오 키워드광고  
푸시 알림  
Daum 검색  
지도

**Response**

Name	Type	Description
result	Result	얼굴 검출 결과를 담는 객체 아래 Result 항목 구성 요소 참고
rid	String	요청에 대한 ID 값

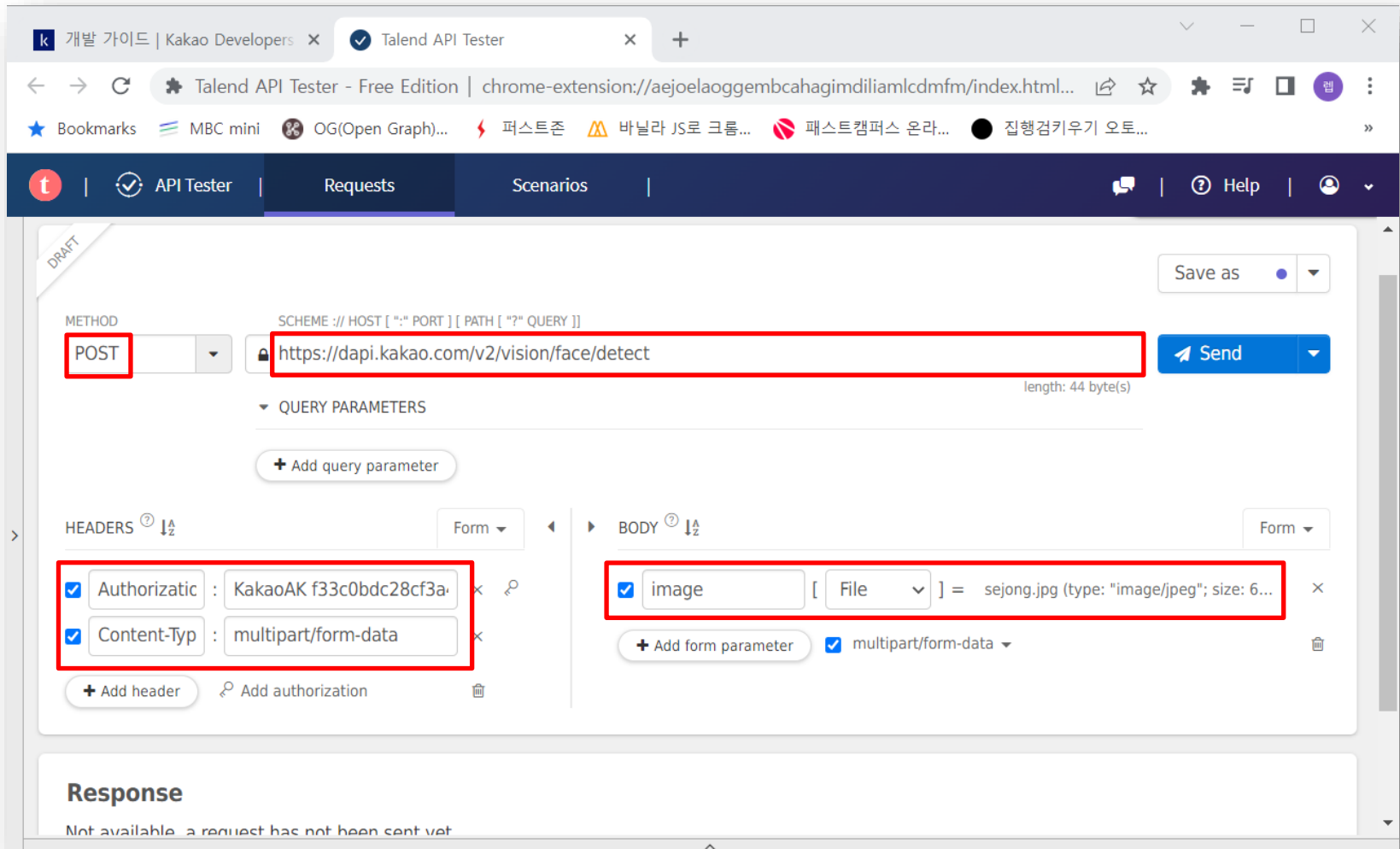
**Result**

Name	Type	Description
width	Integer	요청 이미지의 가로 길이
height	Integer	요청 이미지의 세로 길이
faces	Face[]	검출된 얼굴 정보를 담은 객체의 배열

## ■ Rest API 활용 프로그램 만들기

### ● ex.VisionKakaoAPI

#### 확장프로그램 API Tester



# Rest API 활용 프로그램 만들기

## ● ex.VisionKakaoAPI

### 확장프로그램 API Tester

The screenshot displays the Talend API Tester interface. The top navigation bar includes 'API Tester', 'Requests', and 'Scenarios'. The main area shows a successful response with a status of '200 OK'. The 'HEADERS' section on the left lists various headers such as 'Server: nginx', 'Date: Sun, 19 Jun 2022 00:26:17 GMT', and 'Content-Type: application/json'. The 'BODY' section on the right shows a JSON response, which is highlighted with a red box. The JSON data includes a 'rid' and a 'result' object containing 'width', 'height', 'faces' (an array of facial attributes), and 'age'.

```
1 {
2   rid : "5f190a7289a4b07db84159e8054622b3",
3   result : {
4     width : 530,
5     height : 508,
6     faces : [
7       {
8         facial_attributes : {
9           gender : {
10             male : 0.9999740123748779,
11             female : 2.5955176170100458e-05
12           },
13           age : 25.67610740661621
14         }
15       }
16     ]
17   }
18 }
```

## ■ Rest API 활용 프로그램 만들기

### ● ex.VisionKakaoAPI - File

```
String boundary = "---" + System.currentTimeMillis() + "---";
con.setRequestProperty("Content-Type", "multipart/form-data; boundary=" + boundary);

OutputStream outputStream = con.getOutputStream();
PrintWriter writer = new PrintWriter(
    new OutputStreamWriter(outputStream, "UTF-8"), true);
String LINE_FEED = "\r\n";

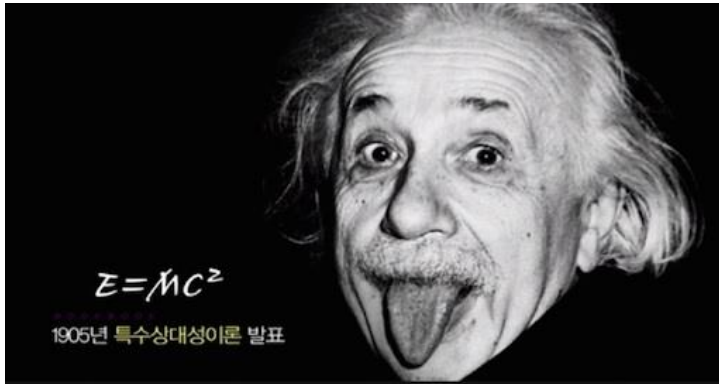
String fileName = uploadFile.getName();
writer.append("--" + boundary).append(LINE_FEED);
writer.append(
    "Content-Disposition: form-data; name=\"image\"; filename=\"" + fileName +
    "\"").append(LINE_FEED);
writer.flush();

FileInputStream inputStream = new FileInputStream(uploadFile);
while (true) {
    int read = inputStream.read();
    if (read == -1) break;
    outputStream.write(read);
}

writer.append(LINE_FEED);
writer.append(boundary + "--").append(LINE_FEED);
writer.close();
```

## ■ Rest API 활용 프로그램 만들기

### ● ex.VisionKakaoAPI - File



-----1655599313743---

Content-Disposition: form-data; name="image";  
filename="einstein.jpg"

2552162552240167470737001110960960025...

---1655599313743-----

```
{
  result: {
    width: 511, height: 271,
    faces: [
      {
        facial_attributes: {
          gender: {
            male: 0.584, female: 0.415
          },
          age: 49.011
        },
        ...
      }
    ]
  }
}
```

## ■ Rest API 활용 프로그램 만들기

### ● ex.VisionKakaoAPI - URL

```
String apiURL = "https://dapi.kakao.com/v2/vision/face/detect";
URL url = new URL(apiURL);
URLConnection con = (URLConnection) url.openConnection();
con.setUseCaches(false);
con.setDoOutput(true);
con.setDoInput(true);

con.setRequestProperty("Authorization", "KakaoAK " + restAPIKey);

OutputStream outputStream = con.getOutputStream();
PrintWriter writer = new PrintWriter(
    new OutputStreamWriter(outputStream, "UTF-8"), true);
writer.append("image_url=" + imageUrl);
writer.flush();
outputStream.flush();
writer.close();

ObjectMapper objectMapper = new ObjectMapper();
Map<String, Object> object =
    objectMapper.readValue(con.getInputStream(), Map.class);
System.out.println(object);
```