

오픈 API를 이용한 번역 어플리케이션 구현 실습

Appendix

- 카카오 개발자 검색 → Kakao developers 회원 가입
- 내 애플리케이션 → 애플리케이션 추가하기



- 문서 → 인공지능 API 가이드 → 번역 → 개발 가이드

카카오 SDK 미지원

번역 API는 REST API 방식으로만 이용할 수 있습니다.

Contents

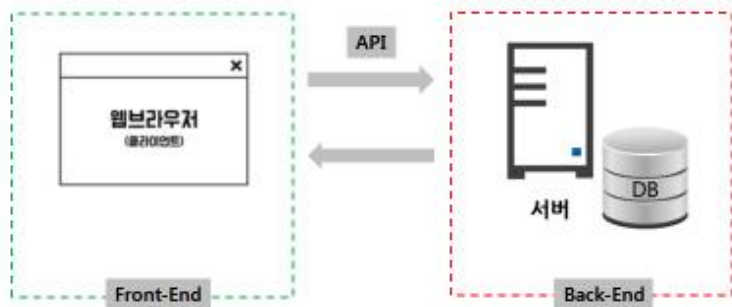
- 용어정리
- API/Open API 개념 및 동작 방식
- Kakao 인공지능 OpenAPI를 이용한 인공지능 SW 구현
 - 번역 어플리케이션

용어 정리

- 전송방식: GET/POST
- JQUERY
 - HTML의 클라이언트 사이드 조작을 단순화 하도록 설계된 **크로스 플랫폼(멀티 플랫폼)의 자바스크립트 라이브러리**
 - JQUERY CDN(Content Delivery Network) 이용
- Ajax(Asynchronous Javascript And Xml)
 - JavaScript를 사용한 비동기 통신, 클라이언트와 서버간에 XML 데이터를 주고받는 기술
- REST API
 - REST는 **HTTP기반으로 필요한 자원에 접근하는 방식**을 정해놓은 아키텍처
 - 자원: DBMS, 이미지/동영상/문서(pdf 등)와 같은 파일, 서비스(이메일 전송, 푸시 메시지 등)를 모두 포함

API/Open API 개념 및 동작 방식

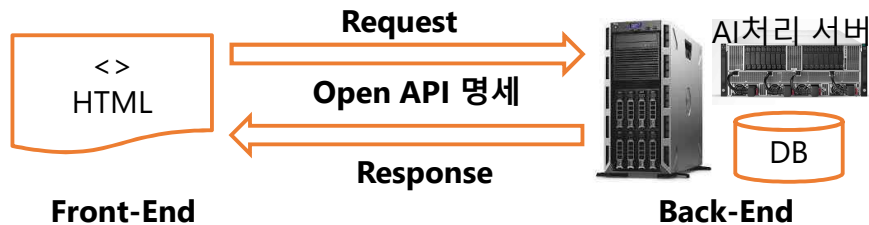
- 웹서비스 개요도



- front-end가 back-end에 요청을 할 때, 특정 규칙[전송방식, Request parameters, etc.]에 맞게 요청
 - 이러한 **사용 규칙을 제공하는 것 = API** (Application Programming Interface)
- Open API: 제공처(국내: 카카오/네이버/ETRI등, 국외: 구글, 페이스북, MS)에서 backend를 만들어 놓고, 그 backend를 이용하는 방법을 제공한다는 의미
 - 사용자는 **backend주소와 사용규칙만 알면, 요청 페이지를 만들어 backend의 자원을 사용 (Serverless)**

API/Open API 개념 및 동작 방식(Example)

- Kakao Open API 명세 (번역)



Open API 명세(Translation)

① Request

- url: <https://dapi.kakao.com/v2/translation/translate>
- 전송방식: GET or POST
- 보낼 것(Parameter)
 - query: 번역할 내용
 - src_lang: 번역 대상 언어
 - target_lang: 번역 결과 언어

Front-End

② Response

- 형식: json(javascript object notation)
- 응답의미 설명
 - translated_text: 번역결과 리스트

Back-End

Request

URL

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

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

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

Key

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

Kakao 인공지능 OpenAPI를 이용한 인공지능 SW 구현

- 번역 어플리케이션

Translate_example x +

← → ↻ ① 파일 | D:/90.Dev.Personal/Dev_openAPI/Example/openAPI_Example.html

Translation Example using OpenAPI

번역 대상 언어
한국어 ▼

번역 결과 언어
영어 ▼

번역할 내용 입력

translation

번역 결과



Translate_example x +

← → ↻ ① 파일 | D:/90.Dev.Personal/Dev_openAPI/Example/openAPI_Example.html

Translation Example using OpenAPI

번역 대상 언어
한국어 ▼

번역 결과 언어
영어 ▼

번역할 내용 입력

오늘은 오픈API를 이용한 인공지능 SW 실습 교육을 진행하도록 하겠습니다.

translation

번역 결과

Today, we will conduct artificial intelligence SW training using open API.

Kakao 인공지능 OpenAPI를 이용한 인공지능 SW 구현

<https://developers.kakao.com/docs/latest/ko/translate/dev-guide>

- 번역 어플리케이션

<https://liveweave.com/>

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Translate_example</title>
</head>
<body>
  <script
    src="https://code.jquery.com/jquery-3.5.1.js"
    integrity="sha256-QWo7LDvxbWT2tbbQ97B53yJnYU3WhH/C8ycbRAKjPDc="
    crossorigin="anonymous"></script>
  <div>
    <h3>Translation Example using OpenAPI</h3>
    <form id="transExForm" name="transExForm">
      번역 대상 언어<br>
      <select id="src_lang" name="src_lang">
        <option value="kr">한국어</option>
        <option value="en">영어</option>
      </select><br>
      번역 결과 언어<br>
      <select id="target_lang" name="target_lang">
        <option value="en">영어</option>
        <option value="kr">한국어</option>
      </select>
      <h3>번역할 내용 입력</h3>
      <textarea id="query" name="query" cols="100" rows="10"></textarea>
    </form>
  </div>
  <div>
    <button onclick="process()">translation</button>
    <h3>번역 결과</h3>
    <textarea id="result_translation" cols="100" rows="10" readonly="true"></textarea>
  </div>
</body>
</html>
```

Front-End 화면 구성 html

```
<script>
function process()
{
  $.ajax({
    url: "https://dapi.kakao.com/v2/translation/translate",
    type: "POST",
    contentType: "application/x-www-form-urlencoded",
    headers: {"Authorization": "KakaoAK 79c427b0a8173417b39d38c9d7e0287d"},
    data: $("#transExForm").serialize(),
    success: function(data)
    {
      //alert("Success");
      $("#result_translation").val(data.translated_text);
    },
    error: function(jqXHR, textStatus, errorThrown)
    {
      var errorMsg = "ready Status: ";
      errorMsg += jqXHR.readyState + "\n";
      errorMsg += "status: ";
      errorMsg += jqXHR.status + "\n";
      errorMsg += "status Text: ";
      errorMsg += jqXHR.statusText + "\n";
      errorMsg += "textStatus: ";
      errorMsg += textStatus + "\n";
      alert(errorMsg);
    }
  });
};
</script>
```

Request/Response 처리 Scripts

REST API 키를 Authorization 헤더에 담아

Kakao 인공지능 OpenAPI를 이용한 인공지능 SW 구현

- Appendix

- JQuery CDN 검색 → uncompressed → script 복사

```
<script
  src="https://code.jquery.com/jquery-3.6.0.js"
  integrity="sha256-H+K7U5CnXl1h5ywQfKtSj8PCmoN9aaq30gDh27Xc0jk="
  crossorigin="anonymous"></script>
```

- Ajax jquery 검색 → [jQuery.ajax\(\) | jQuery API Documentation 페이지](#)

jQuery.ajax(url [settings])

url

Type: [String](#)

A string containing the URL to which the request is sent.

settings

Type: [PlainObject](#)

A set of key/value pairs that configure the Ajax request. All options can be overridden in the [\\$.ajaxSetup\(\)](#) method. See [jQuery.ajax\(settings \)](#) for more details.

error

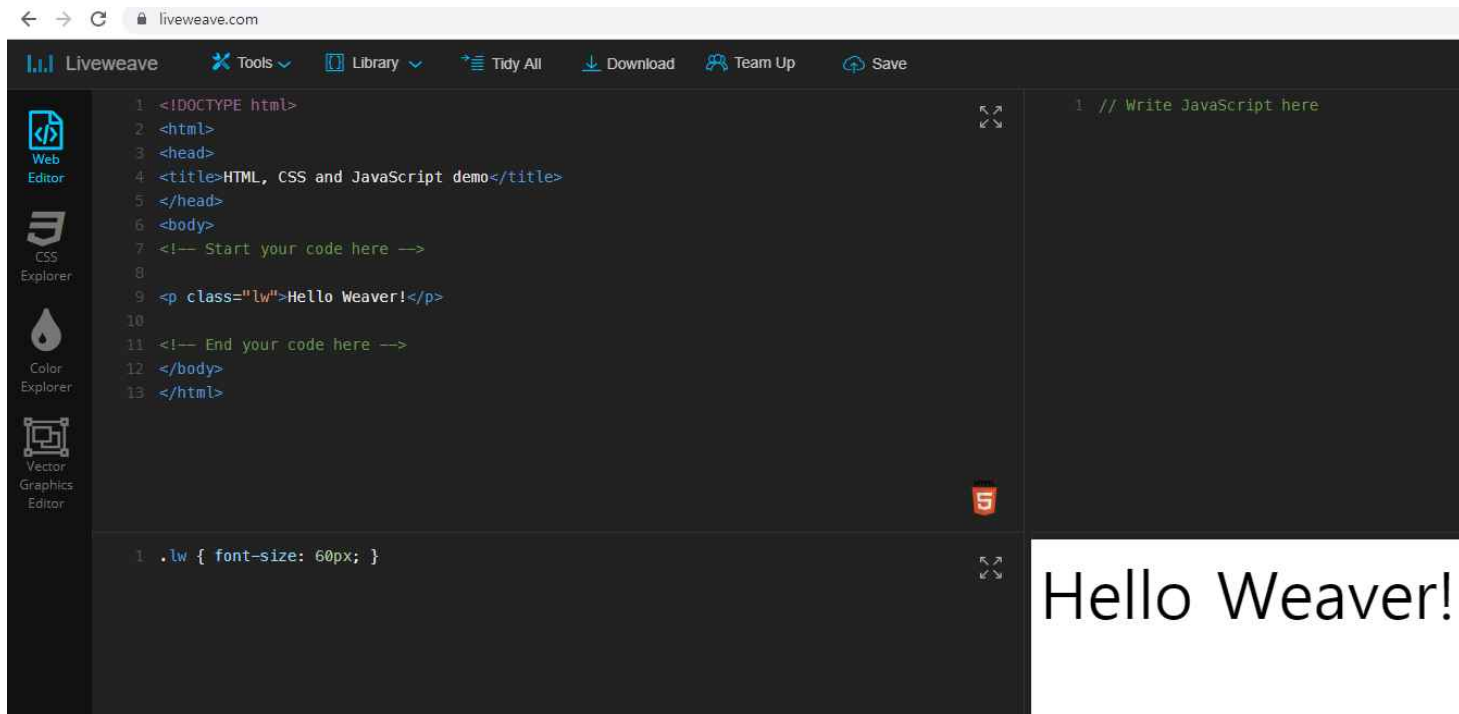
Type: [Function\(jqXHR, String textStatus, String errorThrown \)](#)

A function to be called if the request fails. The function receives three arguments: a [jQuery XMLHttpRequest](#) object, a string describing the type of error that occurred, and an optional exception object, if one occurred. Possible values for the second argument are "timeout", "error", "abort", and "parsererror". When an error is raised by the browser, the [errorThrown](#) argument receives the textual portion of the HTTP status, such as "Server Error." (in HTTP/2 it may instead be an empty string) **As of jQuery 1.9**, the [error](#) callback can accept an array of functions. Each function will be called in turn. **Note** that the [error](#) callback is not called for cross-domain script and cross-domain JSONP requests. This is an [AJAX error](#).

- `readyState`
- `responseXML` and/or `responseText` when the request is successful
- `status`
- `statusText` (may be an empty string in HTTP 1.0)
- `abort([textStatus])`
- `getAllResponseHeaders()` as a string
- `getResponseHeader(name)`
- `overrideMimeType(mimeType)`
- `setRequestHeader(name, value)` which sets the value of the header to the given value rather than concatenating the new value to the existing value
- `statusCode(callbacksByStatusCode)`

Kakao 인공지능 OpenAPI를 이용한 인공지능 SW 구현

- Appendix
 - Liveweave 사용 (HTML 페이지, javascript 작성 용이)



Thank you

Q&A

www.kopo.ac.kr
jsshin7@kopo.ac.kr