

OpenAI. API Reference

- User can interact with the API through HTTP requests from any language.
 - And also there are libraries that OpenAI provided and community provided
 - [Libraries - OpenAI API](#)
 - 맨 아래 부분 보면, Unity Version 도 있음! 또한, [A Non-Official OpenAI Rest Client for Unity \(UPM\) \(github.com\)](#) 이 친구가 Assistant 등의 최신 기능도 만들어 놔서, 더 나은 듯!
 - 자세한 Code Logic 의 경우, 위 Library 를 사용할 것이기 때문에, 자세한 코드 사용의 경우 위 내용을 사용할 수 있도록 하기. Open AI 의 공식 Documentation 을 통해서는, API Endpoint 가 뭐가 존재하는지 + 사용 흐름 (Authentication / 이외 등등...) 에 대해 살펴볼 것임.
-

Get Started

Authentication

- API keys 를 Authentication 에서 사용함.
 - [API Keys](#) 에서 얻을 수 있음.
 - 원칙적으로는 Client Side Code 에서 또한 이 Key 를 공유해서는 안 됨.
 - key 가 보안 및 안전하게 관리될 수 있는 key management service 나, backend server 상에서 관리되는 것이 좋음.
- Q. 그럼 왜 Unity Library 가 존재하나? Unity 는 그냥 Client 밖에 못 만드는데? => A. [해당 Library 링크](#)에서도 다음과 같이 권고함. : source에다가 hard coding 하지 말고, 환경 변수를 통해 load 하도록 사용하는 것이 나음. 적어도, 코드 공유 등의 과정에서 공유되지 않을 수 있도록.
 - OpenAI DotNetProxy를 사용하고 선호하는 OAuth 공급업체를 통해 사용자를 인증하는 것이 훨씬 낫다고도 함!
 - 또한 중간 기여자인 Azure OpenAI 나, OpenAI API Proxy 를 사용할 수도 있음!

Organization

- 여러 organization 에 속해있는 user 라면, 어느 organization 에 해당 API request 를 던질지 고를 수 있음. 그럼 Usage (사용료) 가 해당 organization 에 청구.

Making Request

- HTTP Request 어떻게 구성하냐 / JSON 으로 전달받은 Response 어떻게 처리하냐에 관한 내용임. 이에 대한 내용은 [Web Service & REST & Json / Accessing Web Service](#) 이 두 문서 보면서 좀 더 생각하는 것이 훨 나을 듯.

Streaming

- request 에 대한 partial result 를 stream responses back to a client로 주는 것을 지원함.
 - 이를 위해 [Server-sent events](#) 라는 개념을 사용함.
 - python 및 node.js 에서는 openAI 가 직접 library 에다가 이 기능을 지원하도록 만들어줬음. 근데, 다른 경우에선 내가 *Q. 이 Server-sent events 를 파악하고, 이를 어떻게 활용할 수 있을지 파악하여야 할 듯.*

EndPoints

Embedding, Fine-tuning, Files 는 일단 [OpenAI API Platform 사이트](#)에서도 가능하니까, Chat Completion 과 Assistant 를 기준으로만 살펴보고자 한다!

- EndPoints 는 일반적으로 다음과 같이 제공된다.
 - Request 설명 : HTTP Method / URL / Header / Request Body (Contents)
 - Response 설명 : response 로 반환되는 json object 의 구성.
 - Response 내에 들어있는 추가 Object (Chat Completion / Embedding Object ...) 들에 대한 상세한 설명.

Chat

- Create chat completion : message 와 model 만 제대로 알면 됨.
- response : choices[0] 안의 message 안에 반환된 내용이 일반적으로 들어있음.

Beta

Assistants

Create assistant

- Create an assistant with a model and instructions.

Create assistant file

- Create an assistant file by attaching a [File](#) to an [assistant](#).
- assistant 로 file 을 연결함.
 - file 은 retrieval(검색) / code interpret 를 할 때 효과적으로 사용할 수 있음.
 - File 을 새로 만들고, 이를 Assistant 에 연결하는 역할을 수행.

List assistants

- Returns a list of assistants.

List assistants file

- Returns a list of assistant files.
 - 파일 목록이 따로 존재. 이 파일들을 반환.

Retrieve assistant

- Retrieves an assistant.
- ID 를 path parameter 로 사용하여 assistant 를 검색.

Retrieve assistant file

- Retrieves an AssistantFile.
- Assistant 의 Id 와 File 의 Id 모두를 활용하여 Assistant file 을 검색.

Modity assistant

- assistant_id 를 기반으로 assistant 를 찾고, Request Body 에 입력한 내용을 기반으로 해당 Assistant 를 변경함.

Delete assistant

- Delete an assistant.

Delete assistant file

- Delete an assistant file.

Json Object

- The assistant object / The assistant file object : 이 둘에 대한 설명이 존재하는데, 이는 반환되는 Response 등에 들어있는 Json Object 의 세부적인 내용이 무엇인지에 대

한 설명임.

Thread

Create thread

- Create a thread.

Retrieve thread

- Retrieves a thread.
 - Id 를 기반으로 하여 Thread 를 찾음.

Modify thread

- Modifies a thread.
 - Thread 의 metadata 정보를 수정한다.

Delete thread

- Delete a thread.

The thread object

- Thread Object 가 지니는 JSON 형태에 대한 소개.

Messages

Create message

- Create a message.

List messages

- Returns a list of messages for a given thread.

List message files

- Returns a list of message files.
 - message 에 담겨있는 file 을 의미함. (우리가 gpt 에 message 보낼 때 파일 넣어서 전달할 수 있는 것과 비슷한 이치.)

Retrieve message

- Retrieve a message.
 - path parameter 에 입력하는 thread_id 와 message_id 로 message 를 찾는다.

Retrieve message file

- Retrieves a message file.
 - path parameter 에 입력하는 thread_id 와 message_id 와 file_id 로 message 내부에 들어있는 file 을 찾는다.

Modify message

- Modifies a message.

Json Object

- The message object : Thread 에 보내는 Message 자체에 대한 정보.
- The message file object : message 에 붙어져 있는 file 의 list.

Runs

Run / Run Step 의 구분 :

- Run : "run"은 사용자의 요청을 처리하고 응답을 생성하는 전체 프로세스를 의미
- Run Step : "run step"은 이러한 전체 프로세스 중에서 개별적인 단계를 의미. 모델이 사용자의 요청을 처리하는 과정에서 여러 단계를 거치게 되는데, 이때 각각의 단계를 "run step"이라고 함.

Create run

- Create a run.
 - thread_id / assistant_id 입력하면, run 한다.

Create thread and run

- Create a thread and run it in one request.

List runs

- Create a thread and run it in one request.

List run steps

- Returns a list of run steps belonging to a run.

Retrieve run

- Retrieves a run.
 - `thread_id` 와 `run_id` 를 이용하여 검색함.

Retrieve run step

- Retrieves a run step.

Modify run

- Modifies a run.

Submit tool outputs to run

- When a run has the `status:` `"requires_action"` and `required_action.type` is `submit_tool_outputs`, this endpoint can be used to submit the outputs from the tool calls once they're all completed. All outputs must be submitted in a single request.

Cancel a run

- Cancels a run that is `in_progress`.

The run object

- Represents an execution run on a [thread](#).

The run step object

- Represents a step in execution of a run.