

# OpenAI + Unity. OpenAI Library for Unity in Github

[RageAgainstThePixel/com.openai.unity: A Non-Official OpenAI Rest Client for Unity \(UPM\) \(github.com\)](https://github.com/RageAgainstThePixel/com.openai.unity) : 해당 Library 를 기반으로 함. 여기에 나와있는 내용 중, 내가 사용할 내용에 대해 모두 서술.

- MIT License 이므로, "사용 허용, 제한 없음 / 카피, 수정, 머지, 퍼블리시 가능" 단, 물론 저작권은 해당 인물이 가지고 있음. 사용에 관한 내용은 아무 제약이 없지만, 저작권 자체는 그 사람에게 있다는 뜻.

## Installing (설치)

- Open your Unity project settings
- Add the OpenUPM package registry:
  - Name: **OpenUPM**
  - URL: **<https://package.openupm.com>**
  - Scope(s):
    - **com.openai**
    - **com.utilities**

기존에는 UPM 을 통해 직접적으로 Git Url 을 추가하는 것만 이용하였으나, 해당 package 내부에 추가하여야 하는 세부적인 library 가 많거나 할 경우, 여기 나와 있는 scoped registries 를 이용해 해당 namespace 내부에 들어있는 package 를 **일괄적**으로 Package Manager 에서 표시해줄 수 있다.

"Registry"는 패키지 소스의 위치 를 나타낸다.

- 참고 자료 : [범위 지정 레지스트리 - Unity 매뉴얼 \(unity3d.com\)](https://docs.unity3d.com/Manual/ScopedRegistries.html)

- 일괄적으로 추가된 package 들이 Unity Package Manager 의 항목으로 표시.
- Change the Registry from Unity to **My Registries**
- Add the **OpenAI** package

## Authentication

API Key 를 제공할 수 있는 4가지 방법이 존재함.

1. [Pass keys directly with constructor](#) ⚠

2. [Unity Scriptable Object](#) ⚠
3. [Load key from configuration file](#)
4. [Use System Environment Variables](#)

⚠ 표시가 존재하는 경우, API Key 를 Source 혹은 그에 준하는 방법으로 공유하는 경우를 의미함. API Key 는 웬만하면 Project Folder 내에서는 존재하지 않는 것이 좋음.

## 1. Pass keys directly with constructor ⚠

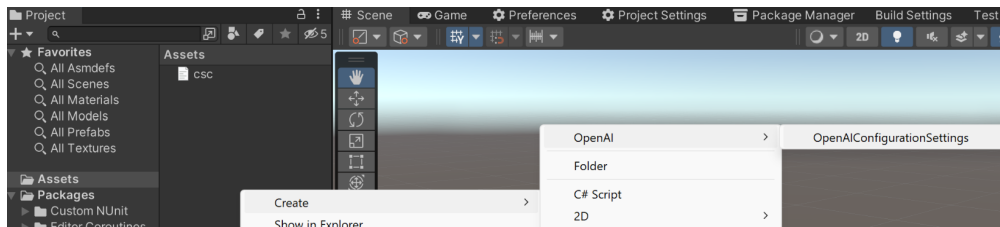
```
var api = new OpenAIClient("sk-apiKey");

var api = new OpenAIClient(new OpenAIAuthentication("sk-apiKey",
"org-yourOrganizationId"));
```

- OpenAIClient 를 다음 인수와 함께 초기화 하는 것으로 충분함.

## 2. Unity Scriptable Object

- Assets/Resources 폴더에, OpenAIConfiguration Scriptable Object 를 입력하여 새로운 API Key 객체를 만들어 저장해둘 수 있음.



- 이것도 위 보안 관련 내용과 동일한 이유로, Source Control 에 해당 파일을 추가하지 않는 것을 권장. OAuth Provider 나, OpenAI-DotNet-Proxy 사용을 권장.

## 3. Load key from configuration file

- user's home directory 를 기준으로, ".openai"로 끝나는 파일을 순회탐색함!
- 해당 파일은 다음과 같은 JSON 구조를 띄면 됨.

```
{
  "apiKey": "sk-aaaabbbbccccddddd",
  "organization": "org-yourOrganizationId"
}
```

- **OpenIAuthentication** 를 이용해 코드 상에서 직접적으로 해당 경로에 접근할 수도 있음.

```
var api = new OpenAIClient(new
OpenIAuthentication().LoadFromDirectory("path/to/your/directory"
));
```

## 4. Use System Environment Variables

- 환경 변수를 이용할 수도 있음.
- Use **OPENAI\_API\_KEY** for your api key.
- Use **OPENAI\_ORGANIZATION\_ID** to specify an organization.

환경 변수란? : 운영 체제에서 프로그램이나 프로세스가 실행될 때 참조할 수 있는 동적으로 할당된 값. Windows 환경 변수 설정에서 해당 내용을 직접 설정해주는 것이 가능하다.

## API Key 를 숨기는 방법

- [OpenAI. API Key 숨기기](#) 에 서술. 해당 내용 참고.