Migrate from OpenAI to Swarms in 3 lines of code

If youve been using GPT-3.5 or GPT-4, switching to Swarms is easy!

Swarms VLMs are available to use through our OpenAI compatible API. Additionally, if you have been building or prototyping using OpenAIs Python SDK you can keep your code as-is and use Swarmss VLMs models.

In this example, we will show you how to change just three lines of code to make your Python application use Swarmss Open Source models through OpenAIs Python SDK.

Getting Started

Migrate OpenAIs Python SDK example script to use Swarmss LLM endpoints.

These are the three modifications necessary to achieve our goal:

Redefine OPENAI_API_KEY your API key environment variable to use your Swarms key.

Redefine OPENAI_BASE_URL to point to `https://api.swarms.world/v1/chat/completions`

Change the model name to an Open Source model, for example: cogvlm-chat-17b

Requirements

We will be using Python and OpenAIs Python SDK.

Instructions

Set up a Python virtual environment. Read Creating Virtual Environments here.

```sh
python3 -m venv .venv
source .venv/bin/activate
```

Install the pip requirements in your local python virtual environment

`python3 -m pip install openai`

Environment setup

To run this example, there are simple steps to take:

Get an Swarms API token by following these instructions.

Expose the token in a new SWARMS_API_TOKEN environment variable:

`export SWARMS_API_TOKEN=<your-token>`

Switch the OpenAI token and base URL environment variable

`export OPENAI_API_KEY=$SWARMS_API_TOKEN`

`export OPENAI_BASE_URL="https://api.swarms.world/v1/chat/completions"`

If you prefer, you can also directly paste your token into the client initialization.

Example code

Once youve completed the steps above, the code below will call Swarms LLMs:

```python
from dotenv import load_dotenv
from openai import OpenAI

load_dotenv()
openai_api_key = ""

openai_api_base = "https://api.swarms.world/v1"
model = "internlm-xcomposer2-4khd"

client = OpenAI(api_key=openai_api_key, base_url=openai_api_base)
# Note that this model expects the image to come before the main text
chat_response = client.chat.completions.create(
    model=model,
    messages=[
        {
            "role": "user",
            "content": [
                {
                    "type": "image_url",
                    "image_url": {
```

"url":

"https://home-cdn.reolink.us/wp-content/uploads/2022/04/010345091648784709.4253.jpg",

},

},

{

"type": "text",

"text": "What is the most dangerous object in the image?",

},

],

}

],

temperature=0.1,

max_tokens=5000,

)

print("Chat response:", chat_response)

```

Note that you need to supply one of Swarmss supported LLMs as an argument, as in the example above. For a complete list of our supported LLMs, check out our REST API page.

## Example output

The code above produces the following object:

```python

ChatCompletionMessage(content="  Hello! How can I assist you today? Do you have any questions or tasks you'd like help with? Please let me know and I'll do my best to assist you.", role='assistant' function_call=None, tool_calls=None)

```
```