```
from swarm_models.openai_function_caller import OpenAIFunctionCaller

from pydantic import BaseModel



# Pydantic is a data validation library that provides data validation and parsing using Python type hints.

# It is used here to define the data structure for making API calls to retrieve weather information.

class WeatherAPI(BaseModel):

    city: str

    date: str



# The WeatherAPI class is a Pydantic BaseModel that represents the data structure

# for making API calls to retrieve weather information. It has two attributes: city and date.



# Example usage:

# Initialize the function caller

function_caller = OpenAIFunctionCaller(

    system_prompt="You are a helpful assistant.",

    max_tokens=500,

    temperature=0.5,

    base_model=WeatherAPI,

)


# The OpenAIFunctionCaller class is used to interact with the OpenAI API and make function calls.
# Here, we initialize an instance of the OpenAIFunctionCaller class with the following parameters:

# - system_prompt: A prompt that sets the context for the conversation with the API.

# - max_tokens: The maximum number of tokens to generate in the API response.

# - temperature: A parameter that controls the randomness of the generated text.

# - base_model: The base model to use for the API calls, in this case, the WeatherAPI class.


# Run the function caller

response = function_caller.run(

    "Get the weather forecast for New York City on July 4th, 2022."

)


# The run() method of the OpenAIFunctionCaller class is used to make a function call to the API.

# It takes a string parameter that represents the user's request or query.

print(response)
```