

```
from swarm_models.openai_function_caller import OpenAIFunctionCaller

from pydantic import BaseModel, Field

from typing import Sequence
```

```
class Page(BaseModel):

    content: str = Field(

        ...,

        description="The content of the page",

    )

    page_number: int = Field(

        ...,

        description="The page number of the page",

    )
```

```
class Chapter(BaseModel):

    title: str = Field(

        ...,

        description="The title of the page",

    )

    page_content: Sequence[Page] = Field(

        ...,

        description="The content of the page in the chapter",

    )
```

Chapter 1 -> chapter 2 -> chapter 3 -> chapter 4 -> chapter 5 -> chapter 6 -> chapter 7 -> chapter 8 -> chapter 9 -> chapter 10

```
class BookSchema(BaseModel):

    book_title: str = Field(

        ...,

        description="The title of the book",

    )

    chapters: Sequence[Chapter] = Field(

        ...,

        description="The chapters of the book",

    )

    conclusion: str = Field(

        ...,

        description="The conclusion of the book",

    )
```

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

```

def generate_book(
    num_chapters: int,
    task: str = "Let's create a fully novel childrens sci fi book with 10 chapters",
):
    for i in range(10):
        responses = []

        responses.append(task)

        model = OpenAIFunctionCaller(
            system_prompt="You're a Book Generator Agent, you're purpose is to generate a fully novel
childrens sci fi book with 10 chapters.",
            max_tokens=3000,
            temperature=1.0,
            base_model=Chapter,
            parallel_tool_calls=False,
        )

        out = model.run(task)
        print(out)
        responses.append(out)

    task = " ".join(responses)

```