Usage Documentation: Discord Bot with Advanced Features

Overview:

This code provides a structure for a Discord bot with advanced features such as voice channel interactions, image generation, and text-based interactions using OpenAI models.

Setup:

1. Ensure that the necessary libraries are installed:

```bash
pip install discord.py python-dotenv dalle3 invoke openai
```

2. Create a `.env` file in the same directory as your bot script and add the following:

```
DISCORD_TOKEN=your_discord_bot_token
STORAGE_SERVICE=your_storage_service_endpoint
SAVE_DIRECTORY=path_to_save_generated_images
```

Bot Class and its Methods:

`__init__(self, agent, llm, command_prefix="!")`:

Initializes the bot with the given agent, language model (`llm`), and a command prefix (default is `!`).

`add_command(self, name, func)`:

Allows you to dynamically add new commands to the bot. The `name` is the command's name and `func` is the function to execute when the command is called.

`run(self)`:

Starts the bot using the `DISCORD_TOKEN` from the `.env` file.

Commands:

1. **!greet**: Greets the user.

2. **!help_me**: Provides a list of commands and their descriptions.

3. **!join**: Joins the voice channel the user is in.

4. **!leave**: Leaves the voice channel the bot is currently in.

5. **!listen**: Starts listening to voice in the current voice channel and records the audio.

6. **!generate_image [prompt]**: Generates images based on the provided prompt using the DALL-E3 model.

7. **!send_text [text] [use_agent=True]**: Sends the provided text to the worker (either the agent or the LLM) and returns the response.

Usage:

Initialize the `llm` (Language Learning Model) with your OpenAI API key:

```python
from swarm_models import OpenAIChat

llm = OpenAIChat(
    openai_api_key="Your_OpenAI_API_Key",
    temperature=0.5,
)
```

Initialize the bot with the `llm`:

```python
from apps.discord import Bot

bot = Bot(llm=llm)
```

Send a task to the bot:

```python
task = "What were the winning Boston Marathon times for the past 5 years (ending in 2022)? Generate a table of the year, name, country of origin, and times."
bot.send_text(task)
```

Start the bot:

```python
bot.run()
```

Additional Notes:

- The bot makes use of the `dalle3` library for image generation. Ensure you have the model and

necessary setup for it.

- For the storage service, you might want to integrate with a cloud service like Google Cloud Storage or AWS S3 to store and retrieve generated images. The given code assumes a method `.upload()` for the storage service to upload files.

- Ensure that you've granted the bot necessary permissions on Discord, especially if you want to use voice channel features.

- Handle API keys and tokens securely. Avoid hardcoding them directly into your code. Use environment variables or secure secret management tools.