

Task

You are tasked with building a **FastAPI** service that exposes endpoints for a simple chatbot. The chatbot should leverage the **OpenAI API** to generate responses. Additionally, the service must store conversation history in PostgreSQL and allow retrieval of this history.

The goal is to test:

1. **Clean coding** and project structure.
2. **API design** using FastAPI.
3. **OpenAI API** usage.
4. **SQL database** integration and schema design.
5. **Best practices** around configuration management, error handling, and security (API keys, etc.).

Requirements

- You must use Python
- You must use OpenAI's API
- You must dockerize your code to be able to run it with only **docker build** and **run**
- You must document how to deploy your code
- Expose at least the following endpoints:
 - **POST /chat**:
 - Accepts a user message and returns a chatbot response.
 - **conversation_id** can be used to group messages under the same conversation.
 - **message** is the user's query.
 - Response
 - **response** is the OpenAI-generated reply.
 - **timestamp** when the message was generated (ISO 8601 format).
 - **GET /history/{conversation_id}**: Returns the entire message history for the given conversation.
 - Response (example):
 - **conversation_id** conversation id.
 - **messages** list of messages in conversation
 - **sender**
 - **text**
 - **timestamp**

Extra points

- **Testing**: Add unit tests or integration tests with **pytest**.
- **Deployment**: Provide **Helmfiles** to deploy de de container.
- **Asynchronous** code with FastAPI (**async def**) where appropriate.