

AnubisGPT Java Chatbot - Detailed Line-by-Line Explanation

This project is a Java Swing-based chatbot UI named AnubisGPT, which integrates with the OpenRouter API (a gateway to several LLMs, like DeepSeek, Mixtral, GPT, etc.), using voice support via FreeTTS.

1. `import javax.swing.*;` -> Imports Java Swing classes for building the UI components.
2. `import java.awt.*;` -> Imports classes for layout and color design.
3. `import java.io.*;` -> Imports classes for file and stream handling.
4. `import java.net.*;` -> Imports classes for URL and HTTP connection handling.
5. `import java.util.ArrayList;` -> Imports the ArrayList class to manage chat history.
6. `import com.google.gson.*;` -> Imports the Gson library for handling JSON data.
7. `import com.sun.speech.freetts.*;` -> Imports FreeTTS classes for text-to-speech.

`public class Anubius extends JFrame { ... }` -> Defines a public class 'Anubius' that extends JFrame to make it a windowed UI app.

Components and Fields:

- JTextArea chatArea: To show the conversation.
- JTextField inputField: Where users type input.
- JButton sendButton, voiceToggleButton: Buttons to send message and toggle voice.
- List<String> messages: To store chat history.
- String OPENROUTER_API_KEY: Your secret API key to access OpenRouter.
- boolean voiceEnabled: Toggles TTS.
- Voice voice: Object from FreeTTS for speaking AI responses.

Constructor Anubius():

- Sets up the main JFrame, layout, sizes, color themes.
- Initializes FreeTTS.
- Creates a top bar with logo and app name.
- Creates the chat area (JTextArea inside JScrollPane).
- Creates the input area with JTextField and Send/Voice buttons.

Voice Handling:

toggleVoice(): Changes voice toggle status and button label.

speak(text): Converts given text to voice using FreeTTS.

Chat Handling:

showGreetingMessage(): Shows welcome text.

sendMessage(): Triggered on Send button or Enter key.

- Appends the users message.
- Starts a new thread to avoid blocking UI.
- Fetches AI response and appends it.
- Optionally speaks it out.

API Integration:

getAIResponse(prompt):

- Sets up POST request to OpenRouter API.
- JSON structure includes model, messages.
- Adds authorization header.
- Parses the response JSON.

- Extracts AI message, cleans it from special characters.

main(): Launches the app using `SwingUtilities.invokeLater`.

Model Used: `deepseek/deepseek-r1-distill-llama-8b` (customizable)

NOTE: FreeTTS works offline, OpenRouter requires internet.