Audio Transcription

Quick Short Project

FlutterFlow

+

OpenAI API

Project Four

Audio Transcription project is a utility tool built with FlutterFlow and powered by the OpenAI Transcription API

Its purpose is to provide immediate and accurate text transcription by converting uploaded audio files (like voice notes or interviews) into clean, readable text, demonstrating simple integration of low-code UI and advanced speech processing

1. Context and Overview

The AI Voice Transcription Assistant is a real-time utility tool that accurately converts spoken audio into text instantly. It integrates FlutterFlow (low-code UI) with the OpenAI Transcription API (supporting formats like mp3, wav, and webm). Users simply upload an audio file and immediately receive a clean, readable text output, showcasing seamless low-code integration for voice-to-text solutions.

2. Inside the OpenAI documentation

Audio Models Overview (Short Explanations)

- Build Voice Agents : AI systems capable of listening, speaking, and responding in real time using natural voice interaction.
- Transcribe Audio: Converts spoken audio into written text with high accuracy. Useful for subtitles, meeting notes, or voice-to-text applications.
- Speak Text :Generates natural-sounding speech from any written text.

API Types (Short Explanations)

Real-Time API
Allows bi-directional audio streams for live conversations (voice agents, assistants, etc.).

Speech API
Converts text back into audio (text-to-speech), enabling "speaking" agents

Transcription API
Used to convert audio files into clean, readable text. This is the API used in this project.

Chat Completion API
Traditional text-based interaction. Useful for turning transcriptions into structured answers, summaries, etc.

Norms & Formats

- **Audio inputs**: mp3, mp4, mpeg, mpga, wav, webm
- **Outputs:** plain text, JSON, SRT (subtitles), VTT (captions)

This is ideal for:

- Voice notes
- Interviews
- Class recordings
- Short audio questions
- Transcribing user messages inside apps

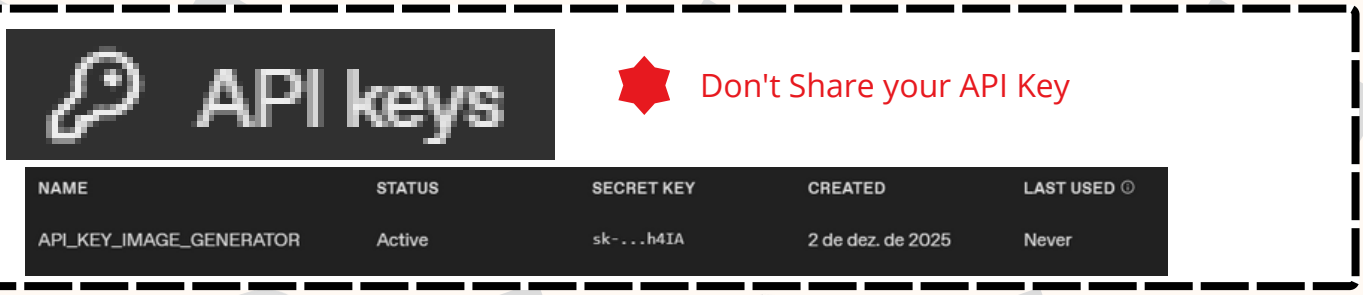
3. Creating the Project and API Keys

In the OpenAI dashboard

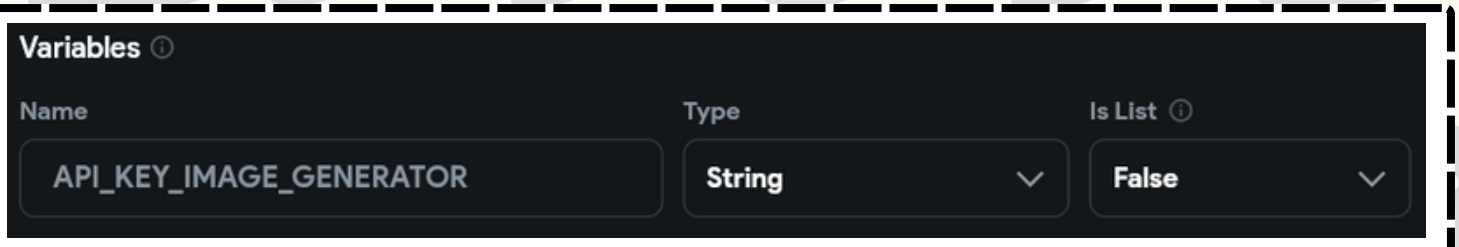
Create the project named

Image Generator

Generate an API Key



Store the key securely (Flutterflow → variables)



4. Applying the Project "Audio Transcription" in FlutterFlow

Container

Title: **Audio Transcription**

Text
Audio Transcription

Button

Send_Audio
Send Audio

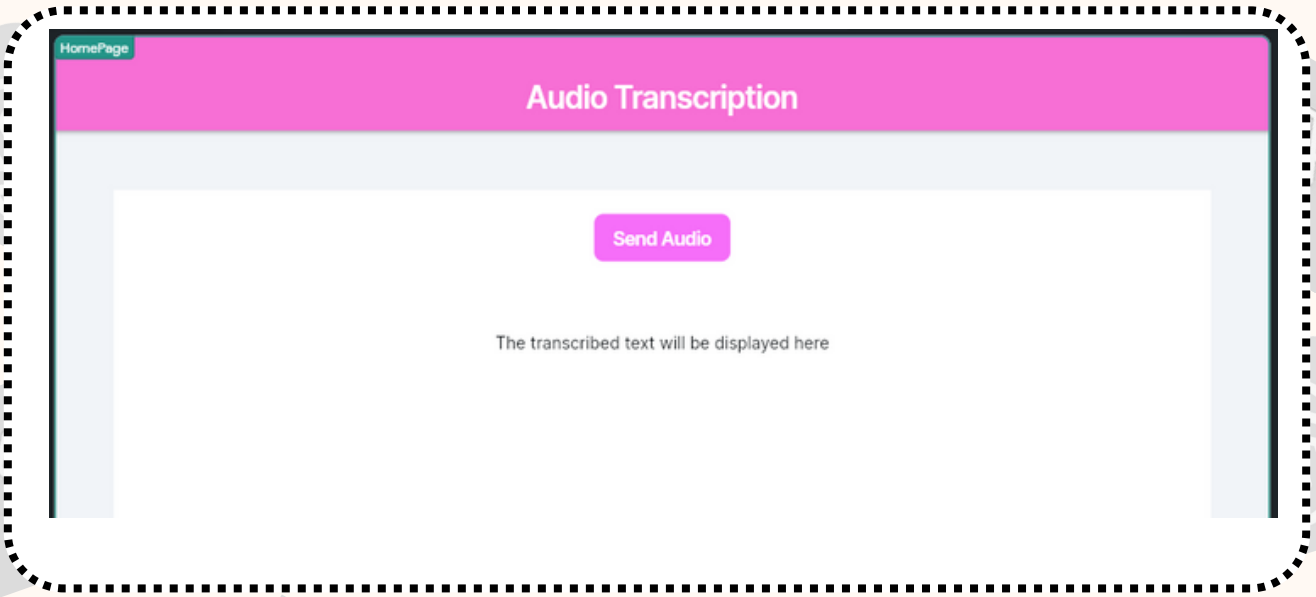
Send_Audio

txt_answer

Answer

txt_answer
The transcribed text will be displayed here

Preview



5. Creating the API Call In FlutterFlow

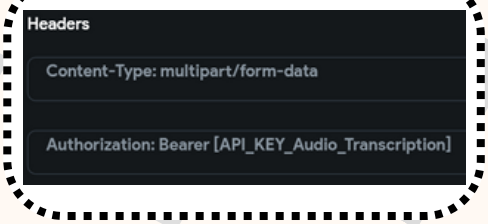
New API Call

Name: "Audio Transcription Call"

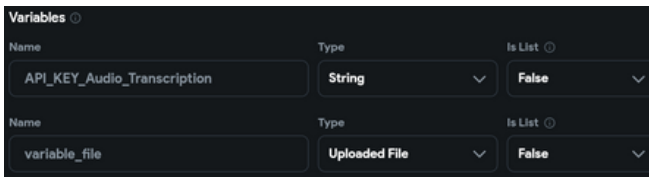
Set as POST

Add headers:

- Content-Type: multipart/form-data
- Authorization: Bearer [API_KEY_Audio_Transcription]



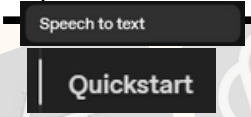
Create variables

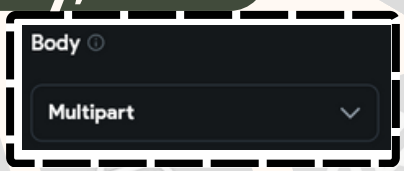


- API_KEY_Audio_Transcription → string
- PROMPT → Uploaded File (input from user)

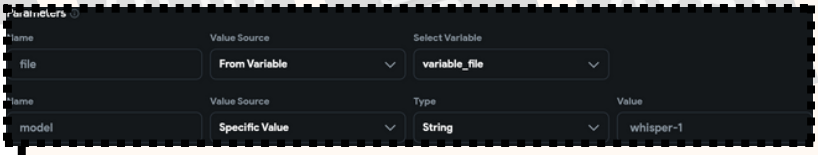
In Multipart Body, add:

the OpenAI API **doc**






```
curl --request POST \
  --url https://api.openai.com/v1/audio/transcriptions \
  --header "Authorization: Bearer $OPENAI_API_KEY" \
  --header 'Content-Type: multipart/form-data' \
  --form file=@/path/to/file/audio.mp3 \
  --form model=gpt-4o-transcribe
```

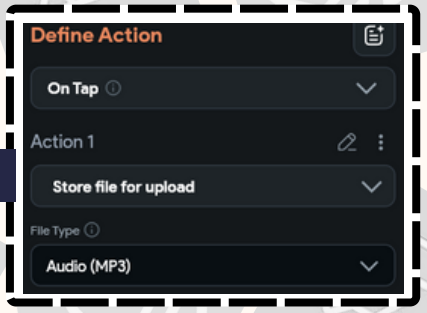


- **Multipart Body: The API Body is set to Multipart, which is the required format for transmitting binary files (like audio) instead of standard JSON text.**
- **Key Parameters: It defines two essential fields: file (linked to variable_file) to pass the dynamic audio content, and model (set as a specific String) to specify the AI engine.**

6. Applying the Image Generator API Inside the App

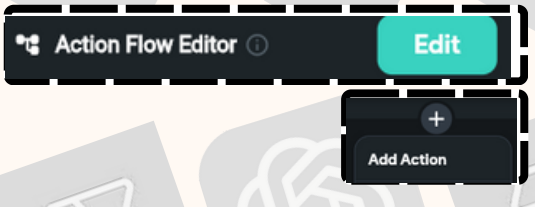
In the "Send Audio" button:

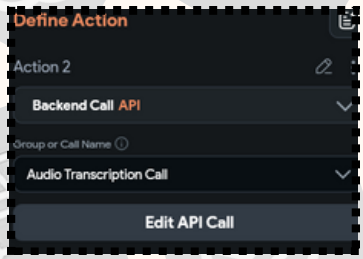
Add Action → **API Call**

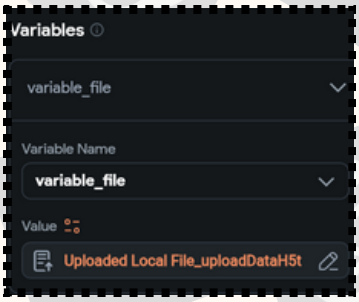


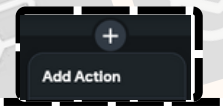
Define Action → Action 1 → Upload Data > **Upload/Save File**

File Type → Audio (MP3)

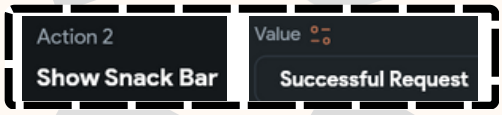








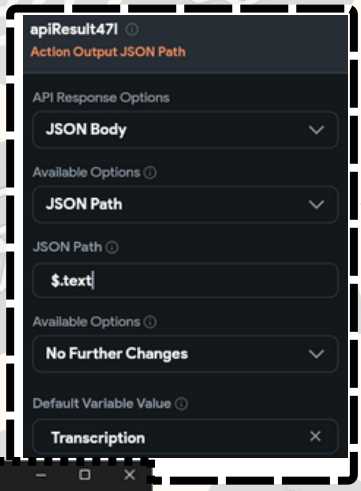
On Success → Show Snackbar **“Successful Request**



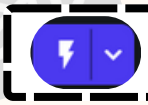
On Error → Show Snackbar **“Error Detected”**

On the answer container:

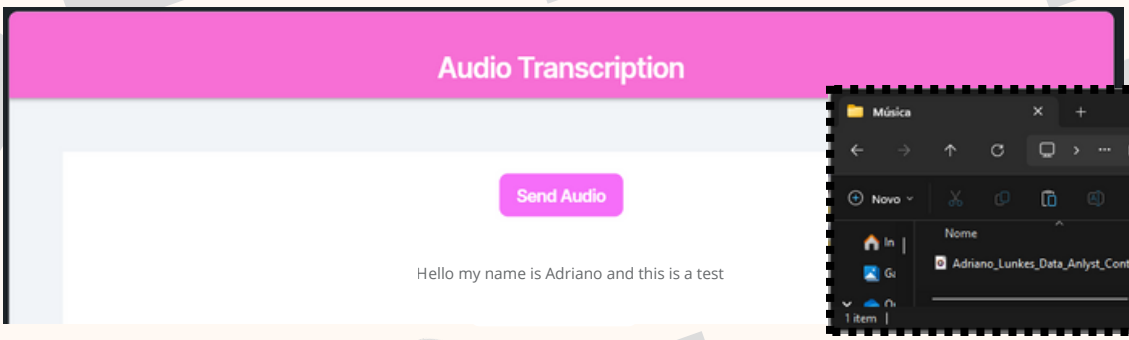
- Set the text to show the API variable **“Result”**



Testing



Send Audio File



7. Conclusion

The Audio Transcription project demonstrates how to

- Use the OpenAI Speech API to convert audio into text
- Upload audio files directly from Flutterflow
- Handle API responses and display transcriptions in a clean interface
- Understand the broader audio ecosystem including real-time, speech, and transcription capabilities

The Audio Transcription Project is a practical, fully functional example of how low-code development (FlutterFlow) + advanced AI (OpenAI Speech API) instantly unlocks the ability to convert spoken audio into accurate text without managing complex backend infrastructure, making sophisticated voice-to-text applications—for interviews, notes, or commands—accessible to anyone in minutes.