**Documentation for Speech-to-Text and Text-to-Speech Integration**

This document provides a detailed explanation of the Python script for recording audio, transcribing it using Google’s Speech-to-Text API, synthesizing speech from text using Google’s Text-to-Speech API, and playing the synthesized audio. The document outlines the modules used, their purpose, rationale, configurations, and sample outputs.

**Table of Contents**

1. [Introduction](https://chatgpt.com/c/677b6a9f-3b5c-800d-9564-40fd2f3127f3#introduction)
2. [Modules Used](https://chatgpt.com/c/677b6a9f-3b5c-800d-9564-40fd2f3127f3#modules-used)
3. [Code Explanation](https://chatgpt.com/c/677b6a9f-3b5c-800d-9564-40fd2f3127f3#code-explanation)
4. [Configuration Details](https://chatgpt.com/c/677b6a9f-3b5c-800d-9564-40fd2f3127f3#configuration-details)
5. [Output Screens](https://chatgpt.com/c/677b6a9f-3b5c-800d-9564-40fd2f3127f3#output-screens)

**Introduction**

This script demonstrates how to:

* Record audio and detect silence to stop recording.
* Transcribe audio using Google’s Speech-to-Text API.
* Convert text to speech using Google’s Text-to-Speech API.
* Play synthesized audio using PyAudio.

**Modules Used**

**1. pyaudio**

* **Purpose**: Handles audio recording and playback.
* **Usage in Script**: Captures audio input and plays audio output.
* **Why Used**: Provides real-time audio capture and playback functionality.

**2. wave**

* **Purpose**: Reads and writes WAV audio files.
* **Usage in Script**: Saves recorded audio to a file and reads audio for playback.
* **Why Used**: Simplifies handling of WAV file formats.

**3. numpy**

* **Purpose**: Provides numerical operations for data processing.
* **Usage in Script**: Processes audio chunks to detect silence.
* **Why Used**: Efficiently handles numerical operations on audio data.

**4. time**

* **Purpose**: Provides time-related functions.
* **Usage in Script**: Detects the duration of silence and tracks transcription time.
* **Why Used**: Required for timing operations.

**5. googleapiclient.discovery**

* **Purpose**: Creates client objects for Google APIs.
* **Usage in Script**: Accesses Speech-to-Text and Text-to-Speech services.
* **Why Used**: Facilitates integration with Google Cloud APIs.

**6. base64**

* **Purpose**: Encodes and decodes data in Base64 format.
* **Usage in Script**: Encodes audio content for API requests and decodes audio content from API responses.
* **Why Used**: Required for transmitting binary data via JSON.

**7. json**

* **Purpose**: Handles JSON data.
* **Usage in Script**: Formats request and response payloads for Google APIs.
* **Why Used**: Required for structured data exchange with APIs.

**8. dotenv**

* **Purpose**: Loads environment variables from a .env file.
* **Usage in Script**: Reads the API key securely.
* **Why Used**: Ensures sensitive information is not hard-coded.

**9. os**

* **Purpose**: Interacts with the operating system.
* **Usage in Script**: Retrieves environment variables.
* **Why Used**: Accesses the API key securely.

**10. gtts**

* **Purpose**: Converts text to speech.
* **Usage in Script**: Synthesizes speech as an alternative to the Google Text-to-Speech API.
* **Why Used**: Provides a fallback for speech synthesis.

**11. playsound**

* **Purpose**: Plays audio files.
* **Usage in Script**: Plays synthesized speech as an alternative to PyAudio.
* **Why Used**: Provides a simple method for audio playback.

**Code Explanation**

**1. Environment Setup**

load\_dotenv()

api\_key = os.getenv("SPEECH\_TO\_TEXT\_API")

print(api\_key)

* Loads the API key from a .env file.

**2. Audio Recording with Silence Detection**

def record\_audio\_with\_silence():

...

* Records audio until silence is detected for a predefined duration (SILENCE\_DURATION).
* Uses numpy to analyze audio chunks for silence.

**3. Speech-to-Text Transcription**

def transcribe\_audio\_google(file\_path, speech\_client):

...

* Encodes audio as Base64 and sends it to Google’s Speech-to-Text API.
* Parses the API response for transcribed text.

**4. Text-to-Speech Synthesis**

def synthesize\_speech(text, client, output\_file=OUTPUT\_FILE):

...

* Sends text to Google’s Text-to-Speech API and saves the audio response as a WAV file.

**5. Audio Playback**

def play\_audio(file\_path):

...

* Reads and plays the synthesized audio file using pyaudio.

**6. Integration Functions**

* **audio\_record()**: Combines recording and transcription.
* **text\_to\_speech(text)**: Combines speech synthesis and playback.

**Configuration Details**

**Key Parameters**

* **RATE**: Audio sample rate (16 kHz for speech applications).
* **CHUNK**: Audio buffer size for real-time processing.
* **SILENCE\_THRESHOLD**: Maximum amplitude for silence detection.
* **SILENCE\_DURATION**: Duration of silence to stop recording.

**Security Considerations**

* API keys are stored securely in a .env file.

**Output Screens**

**Example 1: Recording Audio**

**Input**: Speak into the microphone. **Output**:

Listening... Speak now.

Silence detected. Stopping recording.

Audio recording saved as: temp\_recording.wav

**Example 2: Transcribing Audio**

**Input**: Recorded audio file. **Output**:

Transcribing...

Time taken to transcribe: 2.45 seconds

Transcription: Hello, how are you?

**Example 3: Synthesizing Speech**

**Input**: Text to synthesize: "This is a test." **Output**:

Synthesizing speech...

Audio saved to: output\_audio.wav

Playing audio...

Audio playback finished.

**Summary**

This script provides end-to-end functionality for audio recording, transcription, speech synthesis, and playback. It demonstrates integration with Google Cloud APIs and ensures secure handling of sensitive data.