

# Lab 5

## *It Speaks! Create Synthetic Speech Using Text-to-Speech Speech-to-Text API: Qwik Start*

R Jawahar

## Introduction

This report documents two Qwiklabs exercises from Google Cloud Skills Boost. Each figure is displayed on the left with a concise explanation on the right so that images and steps remain visually aligned and easy to follow.

## 1 It Speaks! — Create Synthetic Speech Using Text-to-Speech (GSP222)

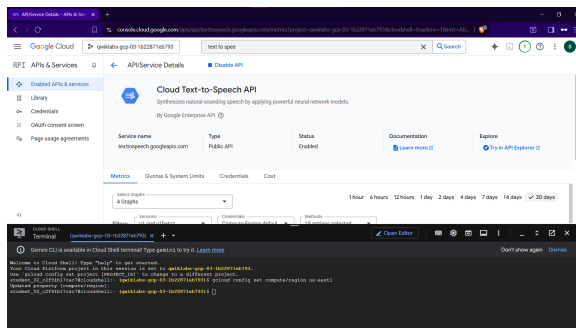


Figure 1: Enable the Text-to-Speech API in Google Cloud Console.

### Explanation:

Enabling the Text-to-Speech API registers the service with your project and allows subsequent API calls. This step is required before creating credentials or sending synthesis requests. In the console you typically click **Enable** or check the API status for your project.

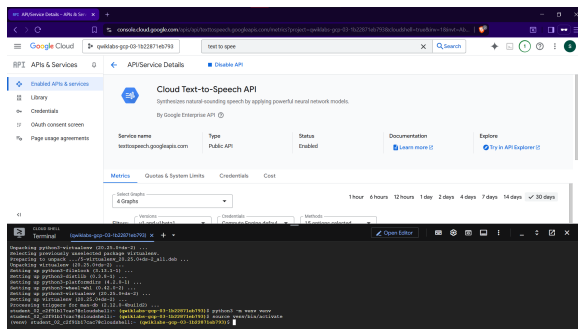


Figure 2: Create a Python virtual environment (venv).

### Explanation:

A virtual environment isolates project dependencies. Typical commands used are:

`python3 -m venv venv` then `source venv/bin/activate`

After activation, install client libraries (e.g., `pip install google-cloud-texttospeech`).

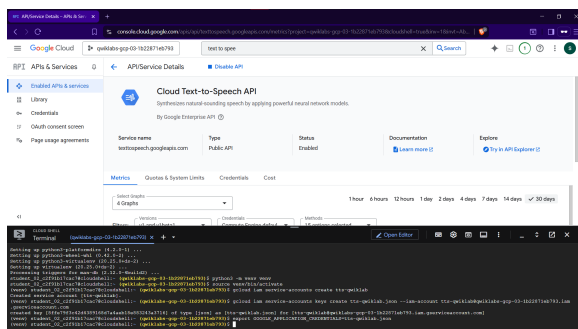


Figure 3: Create a service account and download its key.

### Explanation:

Service accounts provide programmatic authentication. Create one with the required roles (e.g., **Text-to-Speech Client**) and download the JSON key. Set `GOOGLE_APPLICATION_CREDENTIALS` to point to that JSON file for local testing.

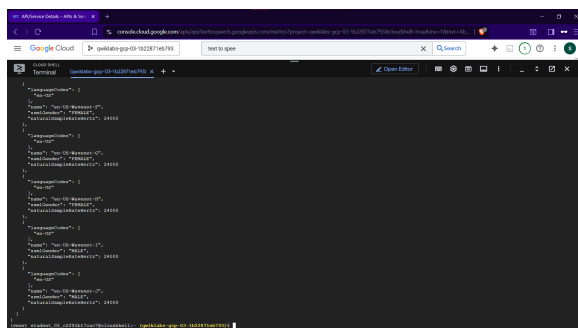


Figure 4: Retrieve the list of available voices.

### Explanation:

Use the client library or a curl request to list voices the API provides (WaveNet and Standard variants). Choose voice name, language code and ssmlGender for your synthesis requests.



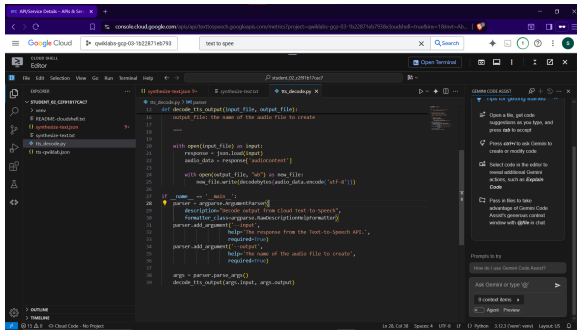


Figure 8: Python script to decode Base64 into an audio file.

### Explanation:

The script reads the JSON response, extracts `audioContent`, decodes with `base64.b64decode()`, and writes bytes to `output.mp3` or `output.wav`. Verify the file plays in a media player.

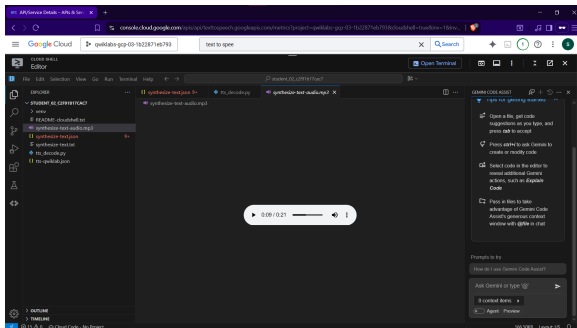


Figure 9: Synthesized audio file created successfully.

### Explanation:

The created audio file contains the spoken version of your input. Listen to confirm pronunciations and adjust voice/SSML settings as needed.

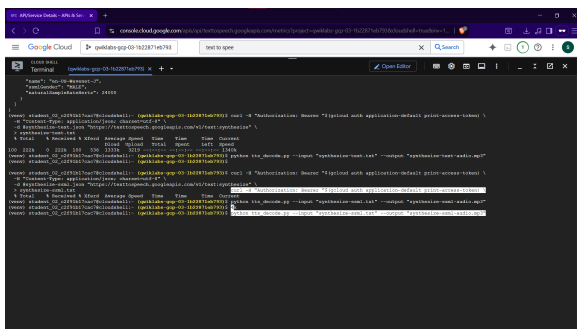


Figure 10: Use SSML to control speech details.

### Explanation:

SSML (Speech Synthesis Markup Language) lets you insert pauses, phonetic hints, emphasis, and prosody tags. Replace the plain text input with an SSML string in the request to fine-tune output.

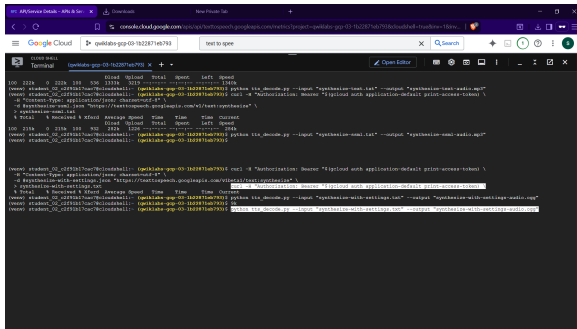


Figure 11: Configure audio output and device profiles.

### Explanation:

Audio configurations include sample rate, pitch, speaking rate, and audio encoding. Device profiles (e.g., headphone) optimize EQ and loudness for target playback devices.

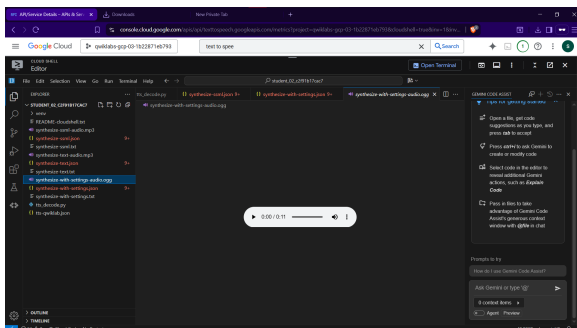


Figure 12: Overview of all generated audio files.

### Explanation:

Compare variants (plain text vs SSML vs device-optimized) to choose the best configuration for your use case. Listen, note differences in prosody and clarity, and document final settings.

## 2 Speech-to-Text API: Qwik Start (GSP119)

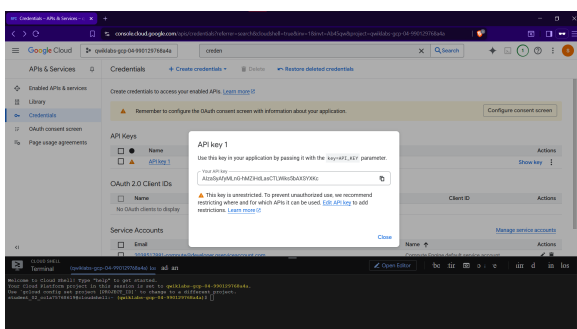


Figure 13: Create an API key and export it as an environment variable.

### Explanation:

For simple tests you can use an API key. For production or broader GCP access, prefer service accounts. When using curl, append `?key=API_KEY` or set appropriate headers.

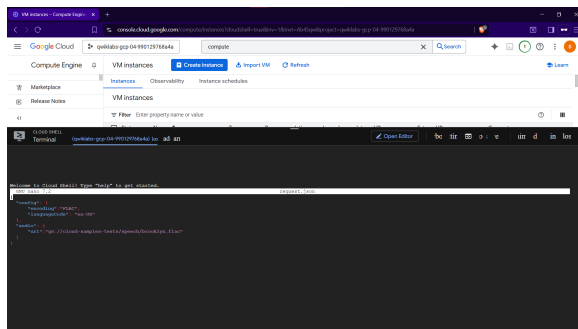


Figure 14: Construct the transcription request (request.json).

### Explanation:

The JSON request references the audio (e.g., `gs://bucket/sample.flac`) and includes recognition config such as `encoding`, `sampleRateHertz`, and `languageCode`. For long audio use asynchronous recognition.

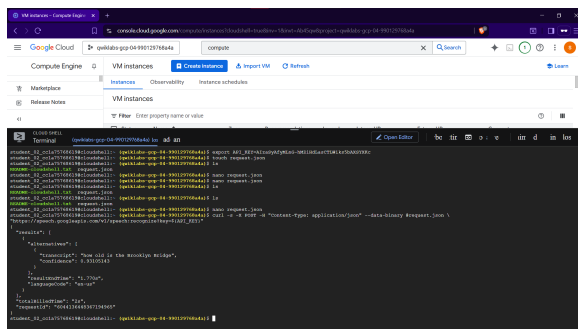


Figure 15: Call the Speech-to-Text API to obtain transcription.

### Explanation:

A curl call returns JSON with transcription results. The top-level field `results[0].alternatives[0].transcript` contains the recognized text. Note the confidence score and, if needed, postprocess punctuation and capitalization.

## Conclusion

All figures are displayed with their explanation aligned to the right for readability. This layout keeps each step and image together, avoids floating far from the related text, and produces a neat, printable report.