## You

I'm writing python code for a text to speech app in a google colab python notebook, here is my code:

Cell 1:

from google.cloud import storage

from google.colab import auth

def uploadAudio(Filepath):

# Authenticate the user

auth.authenticate\_user()

# Initialize the storage client

storage\_client = storage.Client()

# Set your bucket name and file path

bucket\_name = 'speech-to-text-app1' # Replace with your bucket name

audio\_file\_path = Filepath

destination\_blob\_name = Filepath

# Upload the file bucket = storage\_client.bucket(bucket\_name)

blob = bucket.blob(destination\_blob\_name)

blob.upload\_from\_filename(audio\_file\_path)

# Get the GCS URI

gcs\_uri = f'gs://{bucket\_name}/{destination\_blob\_name}'

print(f'File uploaded to {gcs\_uri}')

return gcs\_uri

Cell 2:

!pip install google-cloud-speech

from google.cloud import speech

from google.api\_core.client\_options import ClientOptions

import io

def processAudioFile(audioPath):

# Set the correct project ID

project\_id = "bcft-429709" # Replace with your actual project ID

# Set client options, including the project ID

client\_options = ClientOptions(quota\_project\_id=project\_id)

# Initialize the client with the client options

client = speech.SpeechClient(client\_options=client\_options)

gcs\_uri = uploadAudio(audioPath)

audio = speech.RecognitionAudio(uri=gcs\_uri)

config = speech.RecognitionConfig(

encoding=speech.RecognitionConfig.AudioEncoding.LINEAR16,

sample\_rate\_hertz=16000,

language\_code="en-US",

)

# Use long\_running\_recognize for longer audio files

operation = client.long\_running\_recognize(config=config, audio=audio)

# Wait for the operation to complete

response = operation.result(timeout=900)

print(response)

return response

Cell 3:

# Authenticate the user

from google.colab import auth

auth.authenticate\_user()

from google.colab import files

print("Please upload an audio file")

audio\_upload = files.upload()

# Extract the filename from the uploaded file

audio\_filename = next(iter(audio\_upload.keys())) if audio\_upload else None

if audio\_filename and (audio\_filename.lower().endswith('.mp3') or audio\_filename.lower().endswith('.wav') or audio\_filename.lower().endswith('.m4a')):

audio\_path = audio\_filename

print(f"Audio file uploaded successfully: {audio\_filename}")

else:

print("Error: Please upload a Audio file (.mp3, .wav or .m4a)")

response = processAudioFile(audio\_path)

try:

print("The response is as follows:", response)

# Check if there are any results

if not response.results:

print("No transcription results found.")

else:

for result in response.results:

print("Transcript: {}".format(result.alternatives[0].transcript))

except Exception as e:

print(f"Error during speech recognition: {e}")

I'm running cell 3 directly and the other 2 cells are run indirectly through function calls, however I'm not getting a transcript, this is my output:

Audio file uploaded successfully: (Audio) Screenrecorder-2024-09-02-09-37-06-336.m4a  
File uploaded to gs://speech-to-text-app1/(Audio) Screenrecorder-2024-09-02-09-37-06-336.m4a  
total\_billed\_time {  
seconds: 147  
}  
request\_id: 3443228437626341026

The response is as follows:  
total\_billed\_time {  
seconds: 147  
}  
request\_id: 3443228437626341026

No transcription results found.

Why could this be and how can I fix it, do I have to use a different method maybe?

## Copilot

It looks like your code is correctly uploading the audio file and making a request to the Google Cloud Speech-to-Text API, but the API isn't returning any transcription results. Here are a few things you can check and try to fix the issue:

1. **Audio Encoding and Format**: Ensure that the audio file is in the correct format and encoding. Your code specifies LINEAR16 encoding and a sample rate of 16000 Hz. If your audio file is in a different format, you might need to convert it. You can use tools like ffmpeg to convert your audio file to the required format.

!ffmpeg -i input.m4a -ar 16000 -ac 1 output.wav

1. **Language Code**: Verify that the language code (en-US) matches the language spoken in the audio file. If the audio is in a different language, you need to update the language\_code parameter accordingly.
2. **Audio File Content**: Ensure that the audio file contains clear speech and is not too noisy. Background noise or unclear speech can affect the transcription accuracy.
3. **API Quotas and Limits**: Check if you have reached any API quotas or limits. You can monitor your usage in the Google Cloud Console.
4. **Debugging the Response**: Print out more details from the response to understand why no transcription results are found. For example, you can print the response object to see if there are any error messages or additional information.

print(response)

1. **Alternative Methods**: If the issue persists, you might want to try using the recognize method instead of long\_running\_recognize for shorter audio files. This method is synchronous and might give you more immediate feedback.

response = client.recognize(config=config, audio=audio)

Here's an updated version of your processAudioFile function with additional debugging:

def processAudioFile(audioPath):

# Set the correct project ID

project\_id = "bcft-429709" # Replace with your actual project ID

# Set client options, including the project ID

client\_options = ClientOptions(quota\_project\_id=project\_id)

# Initialize the client with the client options

client = speech.SpeechClient(client\_options=client\_options)

gcs\_uri = uploadAudio(audioPath)

audio = speech.RecognitionAudio(uri=gcs\_uri)

config = speech.RecognitionConfig(

encoding=speech.RecognitionConfig.AudioEncoding.LINEAR16,

sample\_rate\_hertz=16000,

language\_code="en-US",

)

# Use long\_running\_recognize for longer audio files

operation = client.long\_running\_recognize(config=config, audio=audio)

# Wait for the operation to complete

response = operation.result(timeout=900)

# Print the full response for debugging

print(response)

return response

Try these steps and see if they help resolve the issue. Let me know if you need further assistance!

### Learn more:

1. [codelabs.developers.google.com](https://codelabs.developers.google.com/codelabs/cloud-text-speech-python3/)
2. [colab.research.google.com](https://colab.research.google.com/github/r9y9/Colaboratory/blob/master/DeepVoice3_single_speaker_TTS_en_demo.ipynb)
3. [codelabs.developers.google.com](https://codelabs.developers.google.com/codelabs/cloud-speech-text-python3/)
4. [cloud.google.com](https://cloud.google.com/text-to-speech)
5. [cloud.google.com](https://cloud.google.com/text-to-speech/docs)
6. [cloud.google.com](https://cloud.google.com/python)
7. [github.com](https://github.com/googleapis/google-cloud-python)
8. [github.com](http://github.com)
9. [r9y9.github.io](http://r9y9.github.io)
10. [github.com](http://github.com)
11. [www.dropbox.com](http://www.dropbox.com)
12. [www.dropbox.com](http://www.dropbox.com)

## You

How can I find out the audio encoding and format of my audio on the web>?

## Copilot

To find out the audio encoding and format of your audio file, you can use various tools and websites. Here are a few methods:

1. **Online Tools**:
   * Websites like MediaInfo allow you to upload your audio file and get detailed information about its encoding, format, bit rate, sample rate, and more.
2. **Command Line Tools**:
   * If you have access to a command line interface, you can use tools like ffmpeg or sox to get detailed information about your audio file. For example, with ffmpeg, you can use the following command:

ffmpeg -i your\_audio\_file.m4a

* + This will display detailed information about the audio file, including its encoding and format.

1. **Audio Software**:
   * Software like Audacity or VLC Media Player can also provide detailed information about your audio files. In VLC, you can go to Tools > Media Information to see the codec and format details.
2. **Google Cloud Documentation**:
   * Google Cloud's [Introduction to audio encoding for Speech-to-Text](https://cloud.google.com/speech-to-text/docs/encoding) provides detailed information about different audio encodings and formats[^1^](https://cloud.google.com/text-to-speech).

Using these methods, you should be able to determine the encoding and format of your audio file. Let me know if you need any further assistance!

Learn more:

1. [cloud.google.com](https://cloud.google.com/speech-to-text/docs/encoding)
2. [en.wikipedia.org](https://en.wikipedia.org/wiki/Comparison_of_audio_coding_formats)
3. [en.wikipedia.org](https://en.wikipedia.org/wiki/Audio_file_format)
4. [support.serato.com](https://support.serato.com/hc/en-us/articles/202841084-Audio-Encoding-101)

## You

How can I insert into a new word document using python-docx?

## Copilot

To insert content into a new Word document using the python-docx library, you can follow these steps:

1. **Install the python-docx library**: If you haven't already installed the library, you can do so using pip:

pip install python-docx

1. **Create a new Word document and add content**: Here's a basic example of how to create a new Word document and add a heading, a paragraph, and a table:

from docx import Document

from docx.shared import Inches

# Create a new Document

document = Document()

# Add a heading

document.add\_heading('Document Title', level=1)

# Add a paragraph

document.add\_paragraph('This is a paragraph in the document.')

# Add a table

table = document.add\_table(rows=2, cols=2)

table.cell(0, 0).text = 'Cell 1,1'

table.cell(0, 1).text = 'Cell 1,2'

table.cell(1, 0).text = 'Cell 2,1'

table.cell(1, 1).text = 'Cell 2,2'

# Add an image

document.add\_picture('path/to/image.png', width=Inches(1.25))

# Save the document

document.save('example.docx')

1. **Run the script**: Execute the script to generate the Word document with the specified content.

This example demonstrates the basic functionality of the python-docx library. You can add more complex elements like styled text, lists, and more by exploring the [python-docx documentation](https://python-docx.readthedocs.io/en/latest/user/quickstart.html)[^1^](https://cloud.google.com/text-to-speech).

Feel free to ask if you have any specific requirements or need further assistance!

### Learn more:

1. [python-docx.readthedocs.io](https://python-docx.readthedocs.io/)
2. [www.youtube.com](http://www.youtube.com)
3. [pythoninoffice.com](http://pythoninoffice.com)
4. [www.youtube.com](http://www.youtube.com)
5. [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
6. [github.com](https://github.com/python-openxml/python-docx)
7. [python-docx.readthedocs.io](https://python-docx.readthedocs.io/en/latest/user/quickstart.html)
8. [blog.finxter.com](http://blog.finxter.com)
9. [www.geeksforgeeks.org](http://www.geeksforgeeks.org)

## You

I have an audio filename such as myAudio1.mp3 or myAudio2.m4a, how can I replace the bit after the dot with docx using string methods in python?

## Copilot

You can easily replace the file extension in a filename using Python's string methods. Here's a simple way to do it:

# Example filenames

filename1 = "myAudio1.mp3"

filename2 = "myAudio2.m4a"

# Function to replace the extension

def replace\_extension(filename, new\_extension):

# Split the filename into name and extension

name, \_ = filename.rsplit('.', 1)

# Return the new filename with the new extension

return f"{name}.{new\_extension}"

# Replace the extensions

new\_filename1 = replace\_extension(filename1, "docx")

new\_filename2 = replace\_extension(filename2, "docx")

print(new\_filename1) # Output: myAudio1.docx

print(new\_filename2) # Output: myAudio2.docx

This function works by splitting the filename at the last dot (.) and then reconstructing the filename with the new extension. You can use this function to replace any file extension with .docx or any other extension you need.

### Learn more:

1. [stackoverflow.com](https://stackoverflow.com/questions/29413163/rename-a-file-and-only-modify-after-dot)
2. [stackoverflow.com](https://stackoverflow.com/questions/63511184/how-to-replace-multiple-substrings-in-a-filename-while-iterating-through-a-folde)
3. [discuss.python.org](https://discuss.python.org/t/use-of-re-sub-for-renaming-files-strings-not-working/20351)
4. [zetcode.com](https://zetcode.com/python/replace-string/)
5. [en.wikipedia.org](https://en.wikipedia.org/wiki/Comparison_of_audio_coding_formats)
6. [en.wikipedia.org](https://en.wikipedia.org/wiki/Audio_file_format)
7. [support.serato.com](https://support.serato.com/hc/en-us/articles/202841084-Audio-Encoding-101)
8. [cloud.google.com](https://cloud.google.com/speech-to-text/docs/encoding)