Transcribing short audio files

This page demonstrates how to transcribe a short audio file to text using synchronous speech recognition.

*Synchronous speech recognition* returns the recognized text for short audio (less than ~1 minute) in the response as soon as it is processed. To process a speech recognition request for long audio, use [Asynchronous Speech Recognition](https://cloud.google.com/speech-to-text/docs/async-recognize).

Audio content can be sent directly to Cloud Speech-to-Text, or it can process audio content that already resides in Google Cloud Storage. See also the [audio limits](https://cloud.google.com/speech-to-text/quotas) for synchronous speech recognition requests

## **Performing synchronous speech recognition on a local file**

Here is an example of performing synchronous speech recognition on a local audio file:

def transcribe\_file(speech\_file):  
    """Transcribe the given audio file."""  
    from google.cloud import speech  
    from google.cloud.speech import enums  
    from google.cloud.speech import types  
    client = speech.SpeechClient()  
  
    with io.open(speech\_file, 'rb') as audio\_file:  
        content = audio\_file.read()  
  
    audio = types.RecognitionAudio(content=content)  
    config = types.RecognitionConfig(  
        encoding=enums.RecognitionConfig.AudioEncoding.LINEAR16,  
        sample\_rate\_hertz=16000,  
        language\_code='en-US')  
  
    response = client.recognize(config, audio)  
    # Each result is for a consecutive portion of the audio. Iterate through  
    # them to get the transcripts for the entire audio file.  
    for result in response.results:  
        # the first alternative is the most likely one for this portion.  
        Print (u'Transcript: {}'.format (result.alternatives [0].transcript))

## **Performing synchronous speech recognition on a remote file**

For your convenience, Speech-to-Text API can perform synchronous speech recognition directly on an audio file located in Google Cloud Storage, without the need to send the contents of the audio file in the body of your request.

Here is an example of performing synchronous speech recognition on a file located in Cloud Storage:

def transcribe\_gcs(gcs\_uri):  
    """Transcribes the audio file specified by the gcs\_uri."""  
    from google.cloud import speech  
    from google.cloud.speech import enums  
    from google.cloud.speech import types  
    client = speech.SpeechClient()  
  
    audio = types.RecognitionAudio(uri=gcs\_uri)  
    config = types.RecognitionConfig(  
        encoding=enums.RecognitionConfig.AudioEncoding.FLAC,  
        sample\_rate\_hertz=16000,  
        language\_code='en-US')  
  
    response = client.recognize(config, audio)  
    # each result is for a consecutive portion of the audio. Iterate through  
    # them to get the transcripts for the entire audio file.  
    for result in response.results:  
        # The first alternative is the most likely one for this portion.  
        Print (u'Transcript: {}'.format (result.alternatives[0].transcript))