

[AI & Machine Learning Products](https://cloud.google.com/products/machine-learning/) (https://cloud.google.com/products/machine-learning/)

[Cloud Speech-to-Text](https://cloud.google.com/speech-to-text/) (https://cloud.google.com/speech-to-text/)

[Documentation](https://cloud.google.com/speech-to-text/docs/) (https://cloud.google.com/speech-to-text/docs/) [Guides](#)

Speech-to-Text basics

This document is a guide to the basics of using Speech-to-Text. This conceptual guide covers the types of requests you can make to Cloud Speech-to-Text, how to construct those requests, and how to handle their responses. We recommend that all users of Cloud Speech-to-Text read this guide and one of the associated tutorials before diving into the API itself.

Speech requests

Cloud Speech-to-Text has three main methods to perform speech recognition. These are listed below:

- **Synchronous Recognition** (REST and gRPC) sends audio data to the Speech-to-Text API, performs recognition on that data, and returns results after all audio has been processed. Synchronous recognition requests are limited to audio data of 1 minute or less in duration.
- **Asynchronous Recognition** (REST and gRPC) sends audio data to the Speech-to-Text API and initiates a *Long Running Operation*. Using this operation, you can periodically poll for recognition results. Use asynchronous requests for audio data of any duration up to 480 minutes.
- **Streaming Recognition** (gRPC only) performs recognition on audio data provided within a [gRPC bi-directional stream](http://www.grpc.io/docs/guides/concepts.html#bidirectional-streaming-rpc) (http://www.grpc.io/docs/guides/concepts.html#bidirectional-streaming-rpc). Streaming requests are designed for real-time recognition purposes, such as capturing live audio from a microphone. Streaming recognition provides interim results while audio is being captured, allowing result to appear, for example, while a user is still speaking.

Requests contain configuration parameters as well as audio data. The following sections describe these type of recognition requests, the responses they generate, and how to handle those responses in more detail.

Speech-to-Text API recognition

A Speech-to-Text API synchronous recognition request is the simplest method for performing recognition on speech audio data. Cloud Speech-to-Text can process up to 1 minute of speech audio data sent in a synchronous request. After Cloud Speech-to-Text processes and recognizes all of the audio, it returns a response.

A synchronous request is blocking, meaning that Cloud Speech-to-Text must return a response before processing the next request. Cloud Speech-to-Text typically processes audio faster than realtime, processing 30 seconds of audio in 15 seconds on average. In cases of poor audio quality, your recognition request can take significantly longer.

Cloud Speech-to-Text has both REST and gRPC methods for calling Speech-to-Text API synchronous and asynchronous requests. This article demonstrates the REST API because it is simpler to show and explain basic use of the API. However, the basic makeup of a REST or gRPC request is quite similar. [Streaming Recognition Requests \(https://cloud.google.com/speech-to-text/docs/streaming-recognize\)](https://cloud.google.com/speech-to-text/docs/streaming-recognize) are only supported by gRPC.

Synchronous Speech Recognition Requests

A synchronous Speech-to-Text API request consists of a speech recognition configuration, and audio data. A sample request is shown below:

```
{
  "config": {
    "encoding": "LINEAR16",
    "sampleRateHertz": 16000,
    "languageCode": "en-US",
  },
  "audio": {
    "uri": "gs://bucket-name/path_to_audio_file"
  }
}
```

All Speech-to-Text API synchronous recognition requests must include a speech recognition **config** field (of type [RecognitionConfig](https://cloud.google.com/speech-to-text/docs/reference/rest/v1/RecognitionConfig) (https://cloud.google.com/speech-to-text/docs/reference/rest/v1/RecognitionConfig)). A **RecognitionConfig** contains the following sub-fields:

- **encoding** - (required) specifies the encoding scheme of the supplied audio (of type **AudioEncoding**). If you have a choice in codec, prefer a lossless encoding such as **FLAC** or **LINEAR16** for best performance. (For more information, see [Audio Encodings](https://cloud.google.com/speech-to-text/docs/encoding#audio-encodings) (<https://cloud.google.com/speech-to-text/docs/encoding#audio-encodings>.) The **encoding** field is optional for **FLAC** and **WAV** files where the encoding is included in the file header.
- **sampleRateHertz** - (required) specifies the sample rate (in Hertz) of the supplied audio. (For more information on sample rates, see [Sample Rates](#) (#sample-rates) below.) The **sampleRateHertz** field is optional for **FLAC** and **WAV** files where the sample rate is included in the file header.
- **languageCode** - (required) contains the language + region/locale to use for speech recognition of the supplied audio. The language code must be a [BCP-47](https://tools.ietf.org/html/bcp47) (<https://tools.ietf.org/html/bcp47>) identifier. Note that language codes typically consist of primary language tags and secondary region subtags to indicate dialects (for example, 'en' for English and 'US' for the United States in the above example.) (For a list of supported languages, see [Supported Languages](#) (<https://cloud.google.com/speech-to-text/docs/languages>.)
- **maxAlternatives** - (optional, defaults to 1) indicates the number of alternative transcriptions to provide in the response. By default, the Speech-to-Text API provides one primary transcription. If you wish to evaluate different alternatives, set **maxAlternatives** to a higher value. Note that Cloud Speech-to-Text will only return alternatives if the recognizer determines alternatives to be of sufficient quality; in general, alternatives are more appropriate for real-time requests requiring user feedback (for example, voice commands) and therefore are more suited for streaming recognition requests.
- **profanityFilter** - (optional) indicates whether to filter out profane words or phrases. Words filtered out will contain their first letter and asterisks for the remaining characters (e.g. f***). The profanity filter operates on single words, it does not detect abusive or offensive speech that is a phrase or a combination of words.
- **speechContext** - (optional) contains additional contextual information for processing this audio. A context contains the following sub-field:
 - **phrases** - contains a list of words and phrases that provide hints to the speech recognition task. For more information, see the information on [speech adaptation](#) (<https://cloud.google.com/speech-to-text/docs/context-strength>).

Audio is supplied to Cloud Speech-to-Text through the **audio** parameter of type [RecognitionAudio](https://cloud.google.com/speech-to-text/docs/reference/rest/v1/RecognitionAudio) (<https://cloud.google.com/speech-to-text/docs/reference/rest/v1/RecognitionAudio>). The **audio** field contains **either** of the following sub-fields:

- `content` contains the audio to evaluate, embedded within the request. See [Embedding Audio Content](#) (`#embedded-audio`) below for more information. Audio passed directly within this field is limited to 1 minute in duration.
- `uri` contains a URI pointing to the audio content. The file must not be compressed (for example, gzip). Currently, this field must contain a Google Cloud Storage URI (of format `gs://bucket-name/path_to_audio_file`). See [Passing Audio reference by a URI](#) (`#uri-audio`) below.)

More information on these request and response parameters appears below.

Sample rates

You specify the sample rate of your audio in the `sampleRateHertz` field of the request configuration, and it must match the sample rate of the associated audio content or stream. Sample rates between 8000 Hz and 48000 Hz are supported within Cloud Speech-to-Text. The sample rate for a `FLAC` or `WAV` file can be determined from the file header instead of from the `sampleRateHertz` field.

If you have a choice when encoding the source material, capture audio using a sample rate of 16000 Hz. Values lower than this may impair speech recognition accuracy, and higher levels have no appreciable effect on speech recognition quality.

However, if your audio data has already been recorded at an existing sample rate other than 16000 Hz, do not resample your audio to 16000 Hz. Most legacy telephony audio, for example, use sample rates of 8000 Hz, which may give less accurate results. If you must use such audio, provide the audio to the Speech API at its native sample rate.

Languages

Cloud Speech-to-Text's recognition engine supports a variety of languages and dialects. You specify the language (and national or regional dialect) of your audio within the request configuration's `languageCode` field, using a [BCP-47](https://tools.ietf.org/html/bcp47) (<https://tools.ietf.org/html/bcp47>) identifier.

A full list of supported languages, and explanation of BCP-47 identifier tags, is available on the [Language Support](https://cloud.google.com/speech-to-text/docs/languages) (<https://cloud.google.com/speech-to-text/docs/languages>) page.

Time offsets (timestamps)

Cloud Speech-to-Text can include time offset values (timestamps) for the beginning and end of each spoken word that is recognized in the supplied audio. A time offset value represents the amount of time that has elapsed from the beginning of the audio, in increments of 100ms.

Time offsets are especially useful for analyzing longer audio files, where you may need to search for a particular word in the recognized text and locate it (seek) in the original audio. Time offsets are supported for all our recognition methods: `recognize`, `streamingrecognize`, and `longrunningrecognize`.

Time offset values are only included for the first alternative provided in the recognition response.

To include time offsets in the results of your request, set the `enableWordTimeOffsets` parameter to true in your request configuration. For examples using the REST API or the Client Libraries, see [Using Time Offsets \(Timestamps\)](#).

(<https://cloud.google.com/speech-to-text/docs/async-time-offsets>). For example, you can include the `enableWordTimeOffsets` parameter in the request configuration as shown here:

```
{
  "config": {
    "languageCode": "en-US",
    "enableWordTimeOffsets": true
  },
  "audio": {
    "uri": "gs://gcs-test-data/gettysburg.flac"
  }
}
```

The result returned by the Speech-to-Text API will contain time offset values for each recognized word as shown following:

```
{
  "name": "6212202767953098955",
  "metadata": {
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongRunningRecognizeMetadat",
    "progressPercent": 100,
    "startTime": "2017-07-24T10:21:22.013650Z",
    "lastUpdateTime": "2017-07-24T10:21:45.278630Z"
  },
  "done": true,
  "response": {
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongRunningRecognizeRespons
```

```
"results": [  
  {  
    "alternatives": [  
      {  
        "transcript": "Four score and twenty...(etc)...",  
        "confidence": 0.97186122,  
        "words": [  
          {  
            "startTime": "1.300s",  
            "endTime": "1.400s",  
            "word": "Four"  
          },  
          {  
            "startTime": "1.400s",  
            "endTime": "1.600s",  
            "word": "score"  
          },  
          {  
            "startTime": "1.600s",  
            "endTime": "1.600s",  
            "word": "and"  
          },  
          {  
            "startTime": "1.600s",  
            "endTime": "1.900s",  
            "word": "twenty"  
          },  
          ...  
        ]  
      }  
    ]  
  },  
  {  
    "alternatives": [  
      {  
        "transcript": "for score and plenty...(etc)...",  
        "confidence": 0.9041967,  
      }  
    ]  
  }  
]
```

Selecting models

Cloud Speech-to-Text can use one of several machine learning *models* to transcribe your audio file. Google has trained these speech recognition models for specific audio types and sources.

When you send an audio transcription request to Cloud Speech-to-Text, you can improve the results that you receive by specifying the source of the original audio. This allows the Speech-to-Text API to process your audio files using a machine learning model trained to recognize speech audio from that particular type of source.

To specify a model for speech recognition, include the `model` field in the `RecognitionConfig` (<https://cloud.google.com/speech-to-text/docs/reference/rest/v1p1beta1/RecognitionConfig>) object for your request, specifying the model that you want to use.

Cloud Speech-to-Text can use the following types of machine learning models for transcribing your audio files.

Type	Enum constant	Description	Supported languages
Video	<code>video</code>	Use this model for transcribing audio in en-US only video clips or that includes multiple speakers. For best results, provide audio recorded at 16,000Hz or greater sampling rate. Note: This is a premium model that costs more than the standard rate.	
Phone call	<code>phone_call</code>	Use this model for transcribing audio from a phone call. Typically, phone audio is recorded at 8,000Hz sampling rate. Note: The enhanced phone model is a premium model that costs more than the standard rate.	en-US only
Command and search	<code>command_and_search</code>	Use this model for transcribing shorter audio clips. Some examples include voice commands or voice search.	All available languages (https://cloud.google.com/speech-to-text/docs/languages)
Default	<code>default</code>	Use this model if your audio does not fit one of the previously described models. For example, you can use this for long-	All available languages (https://cloud.google.com/speech-to-text/docs/languages)

- Publicly readable (such as our sample audio files)
- Readable by your service account, if using service account authorization.
- Readable by a user account, if using 3-legged OAuth for user account authorization.

More information about managing access to Google Cloud Storage is available at [Creating and Managing Access Control Lists](#)

(<https://cloud.google.com/storage/docs/access-control/create-manage-lists>) in the Google Cloud Storage documentation.

Speech-to-Text API responses

As indicated previously, a synchronous Speech-to-Text API response may take some time to return results, proportional to the length of the supplied audio. Once processed, the API will return a response as shown below:

```
{
  "results": [
    {
      "alternatives": [
        {
          "confidence": 0.98267895,
          "transcript": "how old is the Brooklyn Bridge"
        }
      ]
    }
  ]
}
```

These fields are explained below:

- **results** contains the list of results (of type **SpeechRecognitionResult**) where each result corresponds to a segment of audio (segments of audio are separated by pauses). Each result will consist of one or more of the following fields:
 - **alternatives** contains a list of possible transcriptions, of type **SpeechRecognitionAlternatives**. Whether more than one alternative appears depends both on whether you requested more than one alternative (by setting **maxAlternatives** to a value greater than 1) and on whether Cloud Speech-to-Text

produced alternatives of high enough quality. Each alternative will consist of the following fields:

- **transcript** contains the transcribed text. See [Handling Transcriptions](#) (#transcriptions) below.
- **confidence** contains a value between 0 and 1 indicating how confident Cloud Speech-to-Text is of the given transcription. See [Interpreting Confidence Values](#) (#confidence-values) below.

If no speech from the supplied audio could be recognized, then the returned **results** list will contain no items. Unrecognized speech is commonly the result of very poor-quality audio, or from language code, encoding, or sample rate values that do not match the supplied audio.

The components of this response are explained in the following sections.

Each synchronous Speech-to-Text API response returns a list of results, rather than a single result containing all recognized audio. The list of recognized audio (within the **transcript** elements) will appear in contiguous order.

Selecting alternatives

Each result within a successful synchronous recognition response can contain one or more **alternatives** (if the **maxAlternatives** value for the request is greater than 1). If Cloud Speech-to-Text determines that an alternative has a sufficient [Confidence Value](#) (#confidence-values), then that alternative is included in the response. The first alternative in the response is always the best (most likely) alternative.

Setting **maxAlternatives** to a higher value than 1 does not imply or guarantee that multiple alternatives will be returned. In general, more than one alternative is more appropriate for providing real-time options to users getting results via a [Streaming Recognition Request](#) (#streaming-requests).

Handling transcriptions

Each alternative supplied within the response will contain a **transcript** containing the recognized text. When provided with sequential alternatives, you should concatenate these transcriptions together.

The following Python code iterates over a result list and concatenates the transcriptions together. Note that we take the first alternative (the zeroth) in all cases.

```
response = service_request.execute()
recognized_text = 'Transcribed Text: \n'
for i in range(len(response['results'])):
    recognized_text += response['results'][i]['alternatives'][0]['transcript']
```



Confidence values

The **confidence** value is an estimate between 0.0 and 1.0. It's calculated by aggregating the "likelihood" values assigned to each word in the audio. A higher number indicates an estimated greater likelihood that the individual words were recognized correctly. This field is typically provided only for the top hypothesis, and only for results where `is_final=true`. For example, you may use the **confidence** value to decide whether to show [alternative results](#)

(<https://cloud.google.com/speech-to-text/docs/reference/rest/v1/speech/recognize#speechrecognitionalternative>) to the user or ask for confirmation from the user.

Be aware, however, that the model determines the "best", top-ranked result based on more signals than the **confidence** score alone (such as sentence context). Because of this there are occasional cases where the top result doesn't have the highest confidence score. If you haven't requested multiple alternative results, the single "best" result returned may have a lower confidence value than anticipated. This can occur, for example, in cases where rare words are being used. A word that's rarely used can be assigned a low "likelihood" value even if it's recognized correctly. If the model determines the rare word to be the most likely option based on context, that result is returned at the top even if the result's **confidence** value is lower than alternative options.

Note: Your code should not expect **confidence** as a required field as it is not guaranteed to be accurate, or even set, in any of the results.

Asynchronous Requests and Responses

An asynchronous Speech-to-Text API request to the [LongRunningRecognize](#)

(<https://cloud.google.com/speech-to-text/docs/reference/rest/v1/speech/longrunningrecognize>) method

is identical in form to a [synchronous Speech-to-Text API request](#) (#synchronous-requests). However, instead of returning a response, the asynchronous request will initiate a *Long Running Operation* (of type [Operation](#) (<https://cloud.google.com/speech-to-text/docs/reference/rest/v1/operations#resource-operation>)) and return this operation to the callee immediately.

A typical operation response is shown below:

```
{
  "name": "operation_name",
  "metadata": {
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongRunningRecognizeMetadat",
    "progressPercent": 34,
    "startTime": "2016-08-30T23:26:29.579144Z",
    "lastUpdateTime": "2016-08-30T23:26:29.826903Z"
  }
}
```

Note that no results are yet present. Cloud Speech-to-Text will continue to process the supplied audio and use this operation to store eventual results, which will appear within the `response` field of the operation (of type [LongRunningRecognizeResponse](#) (<https://cloud.google.com/speech-to-text/docs/reference/rest/v1/LongRunningRecognizeResponse>)) upon completion of the request.

A full response after completion of the request appears below:

```
{
  "name": "1268386125834704889",
  "metadata": {
    "lastUpdateTime": "2016-08-31T00:16:32.169Z",
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongrunningRecognizeMetadat",
    "startTime": "2016-08-31T00:16:29.539820Z",
    "progressPercent": 100
  }
  "response": {
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongRunningRecognizeRespons",
    "results": [{
      "alternatives": [{
        "confidence": 0.98267895,
        "transcript": "how old is the Brooklyn Bridge"
      }]
    }],
  },
}
```

```
"done": True,  
}
```

Note that `done` has been set to `True` and that the operation's response contains a set of results of type `SpeechRecognitionResult` (<https://cloud.google.com/speech-to-text/docs/reference/rest/v1/speech/longrunningrecognize#SpeechRecognitionResult>) which is the same type returned by a synchronous Speech-to-Text API recognition request.

By default, an asynchronous REST response will set `done` to `False`, its default value; however, because JSON does not require default values to be present within a field, when testing whether an operation is completed, you should test both that the `done` field is present and that it is set to `True`.

Streaming Speech-to-Text API Recognition Requests

A streaming Speech-to-Text API recognition call is designed for real-time capture and recognition of audio, within a bi-directional stream. Your application can send audio on the request stream, and receive interim and final recognition results on the response stream in real time. Interim results represent the current recognition result for a section of audio, while the final recognition result represents the last, best guess for that section of audio.

Streaming requests

Unlike synchronous and asynchronous calls, in which you send both the configuration and audio within a single request, calling the streaming Speech API requires sending multiple requests. The first `StreamingRecognizeRequest` must contain a configuration of type `StreamingRecognitionConfig` (<https://cloud.google.com/speech-to-text/docs/reference/rpc/google.cloud.speech.v1#google.cloud.speech.v1.StreamingRecognitionConfig>) without any accompanying audio. Subsequent `StreamingRecognizeRequests` sent over the same stream will then consist of consecutive frames of raw audio bytes.

A `StreamingRecognitionConfig` consists of the following fields:

- `config` - (required) contains configuration information for the audio, of type `RecognitionConfig`

(<https://cloud.google.com/speech-to-text/docs/reference/rpc/google.cloud.speech.v1#google.cloud.speech.v1.StreamingRecognitionConfig>)

and is the same as that shown within synchronous and asynchronous requests.

- **single_utterance** - (optional, defaults to **false**) indicates whether this request should automatically end after speech is no longer detected. If set, Cloud Speech-to-Text will detect pauses, silence, or non-speech audio to determine when to end recognition. If not set, the stream will continue to listen and process audio until either the stream is closed directly, or the stream's limit length has been exceeded. Setting **single_utterance** to **true** is useful for processing voice commands.
- **interim_results** - (optional, defaults to **false**) indicates that this stream request should return temporary results that may be refined at a later time (after processing more audio). Interim results will be noted within responses through the setting of **is_final** to **false**.

Streaming responses

Streaming speech recognition results are returned within a series of responses of type

StreamingRecognitionResponse

(<https://cloud.google.com/speech-to-text/docs/reference/rpc/google.cloud.speech.v1#google.cloud.speech.v1.StreamingRecognizeResponse>)

. Such a response consists of the following fields:

- **speechEventType** contains events of type SpeechEventType (<https://cloud.google.com/speech-to-text/docs/reference/rpc/google.cloud.speech.v1#google.cloud.speech.v1.StreamingRecognizeResponse.SpeechEventType>)
 - . The value of these events will indicate when a single utterance has been determined to have been completed. The speech events serve as markers within your stream's response.
- **results** contains the list of results, which may be either interim or final results, of type StreamingRecognitionResult (<https://cloud.google.com/speech-to-text/docs/reference/rpc/google.cloud.speech.v1#google.cloud.speech.v1.StreamingRecognitionResult>)
 - . The **results** list contains following the sub-fields:
 - **alternatives** contains a list of alternative transcriptions.
 - **isFinal** indicates whether the results obtained within this list entry are interim or are final.

- **stability** indicates the volatility of results obtained so far, with **0.0** indicating complete instability while **1.0** indicates complete stability. Note that unlike **confidence**, which estimates whether a transcription is correct, **stability** estimates whether the given partial result may change. If **isFinal** is set to **true**, **stability** will not be set.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (https://developers.google.com/terms/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated January 21, 2020.