Text-to-Speech allows developers to create natural-sounding, synthetic human speech as playable audio. You can use the audio data files you create using Text-to-Speech to power your applications or augment media like videos or audio recordings (in compliance with the Google Cloud Platform Terms of Service (/terms/) including compliance with all applicable law).

Text-to-Speech converts text or Speech Synthesis Markup Language (SSML) input into audio data like MP3 or LINEAR16 (the encoding used in WAV files).

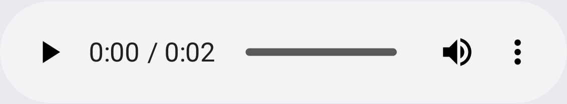
This document is a guide to the fundamental concepts of using Text-to-Speech. Before diving into the API itself, review the quickstarts (/text-to-speech/docs/quickstarts).

Text-to-Speech is ideal for any application that plays audio of human speech to users. It allows you to convert arbitrary strings, words, and sentences into the sound of a person speaking the same things.

Imagine that you have a voice assistant app that provides natural language feedback to your users as playable audio files. Your app might take an action and then provide human speech as feedback to the user.

For example, your app may want to report that it successfully added an event to the user's calendar. Your app constructs a response string to report the success to the user, something like "I've added the event to your calendar."

With Text-to-Speech, you can convert that response string to actual human speech to play back to the user, similar to the example provided below.



Example 1. Audio file generated from Text-to-Speech

To create an audio file like example 1, you send a request to Text-to-Speech like the following code snippet.

The process of translating text input into audio data is called *synthesis* and the output of synthesis is called *synthetic speech*. Text-to-Speech takes two types of input: raw text or SSML-formatted data (discussed below). To create a new audio file, you call the `synthesize` (/text-to-speech/docs/reference/rest/v1/text/synthesize) endpoint of the API.

The speech synthesis process generates raw audio data as a base64-encoded string. You must decode the base64-encoded string into an audio file before an application can play it. Most platforms and operating systems have tools for decoding base64 text into playable media files.

You must decode the base64 string returned from Text-to-Speech before you can play it. For more information about ode base64 data, see Decoding Base64-Encoded Audio Content (/text-to-speech/docs/base64-decoding)

To learn more about synthesis, review the quickstart (/text-to-speech/docs/quickstart) or the Creating Voice Audio Files (/text-to-speech/docs/create-audio) page.

Text-to-Speech creates raw audio data of natural, human speech. That is, it creates audio that sounds like a person talking. When you send a synthesis request to Text-to-Speech, you must specify a *voice* that 'speaks' the words.

Text-to-Speech has a wide selection of custom voices available for you to use. The voices differ by language, gender, and accent (for some languages). For example, you can produce create audio that mimics the sound of a female English speaker with a British accent like example 1, above. You can also convert the same text into a different voice, say a male English speaker with an Australian accent.

▶ 0:00 / 0:01

Example 2. Audio file generated with en-AU speaker

To see the complete list of the available voices, see Supported Voices (/text-to-speech/docs/voices).

Along with other, traditional synthetic voices, Text-to-Speech also provides premium, WaveNet-generated voices. Users find the Wavenet-generated voices to be more warm and human-like than other synthetic voices.

The key difference to a WaveNet voice is the *WaveNet model* used to generate the voice. WaveNet models have been trained using raw audio samples of actual humans speaking. As a result, these models generate synthetic speech with more human-like emphasis and inflection on syllables, phonemes, and words.

Compare the following two samples of synthetic speech.

▶ 0:00 / 0:02

Example 3. Audio file generated with a standard voice

▶ 0:00 / 0:02

Example 4. Audio file generated with a WaveNet voice

To learn more about the benefits of WaveNet-generated voices, see WaveNet and Other Synthetic Voices (/text-to-speech/docs/wavenet).

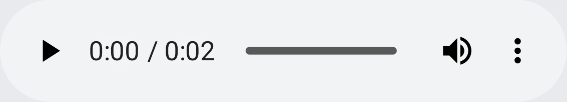
Besides the voice, you can also configure other aspects of the audio data output created by speech synthesis. Text-to-Speech supports configuring the speaking rate, pitch, volume, and sample rate hertz.

Review the AudioConfig reference (/text-to-speech/docs/reference/rest/v1/text/synthesize#audioconfig) for more information.

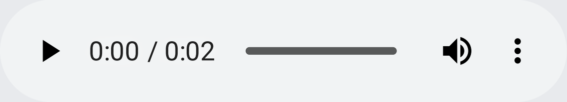
You can enhance the synthetic speech produced by Text-to-Speech by marking up the text using *Speech Synthesis Markup Language (SSML)*. SSML enables you to insert pauses, acronym pronunciations, or other additional details into the audio data created by Text-to-Speech. Text-to-Speech supports a subset of the available SSML elements (/text-to-speech/docs/ssml).

Text-to-Speech does not support all SSML elements for all available languages.

For example, you can ensure that the synthetic speech correctly pronounces ordinal numbers by providing Text-to-Speech with SSML input that marks ordinal numbers as such.



Example 5. Audio file generated from plain text input



Example 6. Audio file generated from SSML input

To learn more about how to synthesize speech from SSML, see Creating Voice Audio Files (/text-to-speech/docs/create-audio#ssml)