AI & Machine Learning Products (https://cloud.google.com/products/machine-learning/)
Cloud Speech-to-Text (https://cloud.google.com/speech-to-text/)
Documentation (https://cloud.google.com/speech-to-text/docs/) Guides

Transcribing audio from streaming input

This section demonstrates how to transcribe streaming audio, like the input from a microphone, to text.

Streaming speech recognition allows you to stream audio to Speech-to-Text and receive a stream speech recognition results in real time as the audio is processed. See also the audio limits (https://cloud.google.com/speech-to-text/quotas) for streaming speech recognition requests. Streaming speech recognition is available via gRPC (https://cloud.google.com/speech-to-text/docs/reference/rpc/google.cloud.speech.v1) only.

Note: To use streaming recognition to stop listening after the user speaks a single word, like in the case of voice commands, set the `single_utterance` field to `true` in the StreamingRecognitionConfig (https://cloud.google.com/speech-to-text/docs/reference/rpc/google.cloud.speech.v1#google.cloud.speech.v1.StreamingRecognitionConfig) object. The `single_utterance` flag tells the Speech API to end the transcription request once it detects that the speech has ended like at the end of a single word.

Performing streaming speech recognition on a local file

The Cloud Speech-to-Text v1 is officially released and is generally available from the `https://speech.googleapis.com/v1/speech` endpoint. The Client Libraries (https://cloud.google.com/speech-to-text/docs/reference/libraries) are released as Alpha and will likely be changed in backward-incompatible ways. The client libraries are currently not recommended for production use.

These samples require that you have set up `gcloud` and have created and activated a service account. For information about setting up `gcloud`, and also creating and activating a service account, see Quickstart (https://cloud.google.com/speech-to-text/docs/quickstart).

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

C# GO JAVA NODE.JS MORE

CLOUDPLATFORM/DOTNET-DOCS-SAMPLES/BLOB/MASTER/SPEECH/API/RECOGNIZE/RECOGNIZE.CS)

FEEDBACK (#)

```
static async Task<object> StreamingRecognizeAsync(string filePath)
{
    var speech = SpeechClient.Create();
    var streamingCall = speech.StreamingRecognize();
    // Write the initial request with the config.
    await streamingCall.WriteAsync(
        new StreamingRecognizeRequest()
        {
            StreamingConfig = new StreamingRecognitionConfig()
            {
                Config = new RecognitionConfig()
                {
                    Encoding =
                    RecognitionConfig.Types.AudioEncoding.Linear16,
                    SampleRateHertz = 16000,
                    LanguageCode = "en",
                },
                InterimResults = true,
            }
        });
    // Print responses as they arrive.
    Task printResponses = Task.Run(async () =>
    {
        while (await streamingCall.ResponseStream.MoveNext(
            default(CancellationToken)))
        {
            foreach (var result in streamingCall.ResponseStream
                .Current.Results)
            {
                foreach (var alternative in result.Alternatives)
                {
                    Console.WriteLine(alternative.Transcript);
                }
            }
        }
    });
    // Stream the file content to the API.  Write 2 32kb chunks per
```

```
    // second.
    using (FileStream fileStream = new FileStream(filePath, FileMode.Open))
    {
        var buffer = new byte[32 * 1024];
        int bytesRead;
        while ((bytesRead = await fileStream.ReadAsync(
            buffer, 0, buffer.Length)) > 0)
        {
            await streamingCall.WriteAsync(
                new StreamingRecognizeRequest()
                {
                    AudioContent = Google.Protobuf.ByteString
                    .CopyFrom(buffer, 0, bytesRead),
                });
            await Task.Delay(500);
        };
    }
    await streamingCall.WriteCompleteAsync();
    await printResponses;
    return 0;
}
```

While you can stream a local audio file to the Speech-to-Text API, it is recommended that you perform synchronous (https://cloud.google.com/speech-to-text/docs/sync-recognize) or asynchronous (https://cloud.google.com/speech-to-text/docs/async-recognize) audio recognition for batch mode results.

Performing streaming speech recognition on an audio stream

The Cloud Speech-to-Text v1 is officially released and is generally available from the **`https://speech.googleapis.com/v1/speech`** endpoint. The Client Libraries (https://cloud.google.com/speech-to-text/docs/reference/libraries) are released as Alpha and will likely be changed in backward-incompatible ways. The client libraries are currently not recommended for production use.

These samples require that you have set up **`gcloud`** and have created and activated a service account. For information about setting up **`gcloud`**, and also creating and activating a service account, see Quickstart (https://cloud.google.com/speech-to-text/docs/quickstart).

Cloud Speech-to-Text can also perform recognition on streaming, real-time audio.

Here is an example of performing streaming speech recognition on an audio stream received from a microphone:

C# GO JAVA NODE.JS PYTHON

CLOUDPLATFORM/DOTNET-DOCS-SAMPLES/BLOB/MASTER/SPEECH/API/RECOGNIZE/RECOGNIZE.CS)

FEEDBACK (#)

```
static async Task<object> StreamingMicRecognizeAsync(int seconds)
{
    var speech = SpeechClient.Create();
    var streamingCall = speech.StreamingRecognize();
    // Write the initial request with the config.
    await streamingCall.WriteAsync(
        new StreamingRecognizeRequest()
        {
            StreamingConfig = new StreamingRecognitionConfig()
            {
                Config = new RecognitionConfig()
                {
                    Encoding =
                    RecognitionConfig.Types.AudioEncoding.Linear16,
                    SampleRateHertz = 16000,
                    LanguageCode = "en",
                },
                InterimResults = true,
            }
        });
    // Print responses as they arrive.
    Task printResponses = Task.Run(async () =>
    {
        while (await streamingCall.ResponseStream.MoveNext(
            default(CancellationToken)))
        {
            foreach (var result in streamingCall.ResponseStream
                .Current.Results)
            {
                foreach (var alternative in result.Alternatives)
                {
                    Console.WriteLine(alternative.Transcript);
```

```
                }
            }
        }
    });
    // Read from the microphone and stream to API.
    object writeLock = new object();
    bool writeMore = true;
    var waveIn = new NAudio.Wave.WaveInEvent();
    waveIn.DeviceNumber = 0;
    waveIn.WaveFormat = new NAudio.Wave.WaveFormat(16000, 1);
    waveIn.DataAvailable +=
        (object sender, NAudio.Wave.WaveInEventArgs args) =>
        {
            lock (writeLock)
            {
                if (!writeMore)
                {
                    return;
                }

                streamingCall.WriteAsync(
                    new StreamingRecognizeRequest()
                    {
                        AudioContent = Google.Protobuf.ByteString
                            .CopyFrom(args.Buffer, 0, args.BytesRecorded)
                    }).Wait();
            }
        };
    waveIn.StartRecording();
    Console.WriteLine("Speak now.");
    await Task.Delay(TimeSpan.FromSeconds(seconds));
    // Stop recording and shut down.
    waveIn.StopRecording();
    lock (writeLock)
    {
        writeMore = false;
    }

    await streamingCall.WriteCompleteAsync();
    await printResponses;
    return 0;
}
```

What's next

- Learn how to transcribe an audio stream endlessly (https://cloud.google.com/speech-to-text/docs/endless-streaming-tutorial)

Last updated January 21, 2020.