
Amazon Transcribe

Developer Guide



Amazon Transcribe: Developer Guide

Copyright © 2019 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What Is Amazon Transcribe?	1
Recognizing Voices	1
Transcribing Separate Audio Channels	2
Transcribing Streaming Audio	2
Custom Vocabulary	2
Are You a First-time User of Amazon Transcribe ?	2
How It Works	3
Speech Input	3
Identifying Speakers	4
Transcribing Streaming Audio	4
Channel Identification	4
Custom Vocabularies	6
Create a Custom Vocabulary	6
English Character Set	7
French Character Set	7
Italian Character Set	8
Portuguese Character Set	9
Spanish Character Set	10
Getting Started	11
Step 1: Set Up an Account	11
Sign up for AWS	11
Create an IAM User	12
Next Step	12
Step 2: Set Up the AWS CLI	12
Next Step	13
Step 3: Getting Started Using the Console	13
Create a Transcription Job	13
View a Transcription Job	14
Step 4: Getting Started Using the API	17
Getting Started (AWS CLI)	17
Getting Started (SDK for Python)	19
Step 5: Getting Started With Streaming Audio	21
Streaming Transcription	25
Using Amazon Transcribe Streaming	25
A Streaming Retry Client	27
Streaming Retry Client Code	27
Streaming Retry Client Interface Code	30
Using the Retry Client	32
Monitoring	34
Monitoring Amazon Transcribe with CloudTrail	34
Amazon Transcribe Information in CloudTrail	34
.....	35
CloudWatch Events	36
Defining CloudWatch Events Rules	37
Amazon Transcribe Event	37
Authentication and Access Control	39
Authentication	39
Access Control	40
Overview of Managing Access	40
Managing Access to Actions	41
Specifying Policy Elements: Resources, Actions, Effects, and Principals	41
Specifying Conditions in a Policy	42
Using Identity-based Policies (IAM Policies) for Amazon Transcribe	42
Permissions Required to Use the Amazon Transcribe Console	43

AWS Managed (Predefined) Policies for Amazon Transcribe	43
Permissions Required for IAM User Roles	44
Permissions Required for Amazon S3 Encryption Keys	44
Amazon Transcribe API Permissions Reference	45
Guidelines and Limits	46
Supported Regions	46
Throttling	46
Guidelines	46
Limits	46
Document History	47
API Reference	48
Actions	48
Amazon Transcribe Service	48
Amazon Transcribe Streaming Service	73
Data Types	77
Amazon Transcribe Service	78
Amazon Transcribe Streaming Service	87
Common Errors	97
Common Parameters	98
AWS Glossary	101

What Is Amazon Transcribe?

Amazon Transcribe uses advanced machine learning technologies to recognize speech in audio files and transcribe them into text. You can use Amazon Transcribe to convert audio to text and to create applications that incorporate the content of audio files. For example, you can transcribe the audio track from a video recording to create closed captioning for the video.

You can use Amazon Transcribe to transcribe the following languages to text:

- Australian English (en-AU)
- British English (en-GB)
- US English (en-US)
- French (fr-FR)
- Canadian French (fr-CA)
- Italian (it-IT)
- Brazilian Portuguese (pt-BR)
- US Spanish (es-US)

You can use Amazon Transcribe with other AWS services to create applications. For example, you can:

- Use Amazon Transcribe to convert voice to text, send the text to Amazon Translate to translate it into another language, and send the translated text to Amazon Polly to speak the translated text.
- Use Amazon Transcribe to transcribe recordings of customer service calls for analysis. After transcribing a recording, send the transcription to Amazon Comprehend to identify keywords, topics, or sentiments.
- Use Amazon Transcribe to transcribe live broadcasts such as television to provide real-time subtitles.

To use Amazon Transcribe you store your audio file in an Amazon S3 bucket. The output from the transcription job is also stored in an S3 bucket. Content delivered to Amazon S3 buckets might contain customer content. For more information about removing sensitive data, see [How Do I Empty an S3 Bucket?](#) or [How Do I Delete an S3 Bucket?](#).

Recognizing Voices

Amazon Transcribe can identify the individual speakers in an audio clip, a technique known as *diarization* or *speaker identification*. When you activate speaker identification, Amazon Transcribe includes an attribute that identifies each speaker in the audio clip. You can use speaker identification to:

- identify the customer and the support representative in a recorded customer support call
- identify characters for closed captions
- identify the speaker and questioners in a recorded press conference or lecture

You can specify the number of voices that you want Amazon Transcribe to recognize in an audio clip

Transcribing Separate Audio Channels

To create a transcript for each channel, or single stream of recorded sound, in an audio file, use *channel identification*. With channel identification, Amazon Transcribe returns two or more transcriptions: a combined transcription of all of the audio channels and a transcription of each audio channel.

Use channel identification when your audio is on multiple channels. For example, use channel identification:

- When your recording has a customer service representative on one channel and a customer on another
- When you transcribe a podcast where the host is recorded on one channel and the guest on another

For more information about channel identification, see [Channel Identification \(p. 4\)](#).

Transcribing Streaming Audio

You can use Amazon Transcribe to transcribe streaming audio in real-time. You send Amazon Transcribe a stream of audio and Amazon Transcribe returns a stream of JSON objects containing the transcription of the audio.

For more information about processing audio streams, see [Streaming Transcription \(p. 25\)](#).

Custom Vocabulary

Create a custom vocabulary to help Amazon Transcribe recognize words that are specific to your use case and improve its accuracy in converting speech to text. For example, you might create a custom vocabulary that includes industry-specific words and phrases.

Use a custom vocabulary to help Amazon Transcribe recognize:

- words that are not being recognized
- unfamiliar words that are specific to your domain

For more information about creating a custom vocabulary, see [Custom Vocabularies \(p. 6\)](#).

Are You a First-time User of Amazon Transcribe ?

If you are a first-time user, we recommend that you read the following sections in order:

1. [How Amazon Transcribe Works \(p. 3\)](#)—Introduces Amazon Transcribe.
2. [Getting Started with Amazon Transcribe \(p. 11\)](#)—Explains how to set up your AWS account and use Amazon Transcribe.
3. [API Reference \(p. 48\)](#)—Contains reference documentation for Amazon Transcribe operations.

How Amazon Transcribe Works

Amazon Transcribe analyzes audio files that contain speech and uses advanced machine learning techniques to transcribe the voice data into text. You can then use the transcription as you would any text document.

To transcribe an audio file, Amazon Transcribe uses three operations:

- [StartTranscriptionJob \(p. 67\)](#) – Starts an asynchronous job to transcribe the speech in an audio file to text.
- [ListTranscriptionJobs \(p. 61\)](#) – Returns a list of transcription jobs that have been started. You can specify the status of the jobs that you want the operation to return. For example, you can get a list of all pending jobs, or a list of completed jobs.
- [GetTranscriptionJob \(p. 56\)](#) – Returns the result of a transcription job. The response contains a link to a JSON file containing the results.

To transcribe streaming audio to text, Amazon Transcribe provides one operation:

- [StartStreamTranscription \(p. 74\)](#) – Starts a bi-directional HTTP/2 stream where audio is streamed to Amazon Transcribe and the transcription results are streamed to your application.

You can also use the Amazon Transcribe to create and manage custom vocabularies for your solution. A custom vocabulary gives Amazon Transcribe more information about how to process speech in an audio clip.

- [CreateVocabulary \(p. 49\)](#) – Creates a custom vocabulary that you can use in your transcription jobs.
- [DeleteVocabulary \(p. 54\)](#) – Deletes a custom vocabulary from your account.
- [GetVocabulary \(p. 58\)](#) – Gets information about a custom vocabulary and a URL that you can use to download the contents of a vocabulary.
- [ListVocabularies \(p. 64\)](#) – Gets a list of custom vocabularies in your account.
- [UpdateVocabulary \(p. 71\)](#) – Updates an existing vocabulary.

You can transcribe speech in any of the following languages:

- Australian English (en-AU)
- British English (en-GB)
- US English (en-US)
- French (fr-FR)
- Canadian French (fr-CA)
- Italian (it-IT)
- Brazilian Portuguese (pt-BR)
- US Spanish (es-US)

Speech Input

To transcribe an audio file, you use a transcription job. You store the file as an object in an Amazon S3 bucket. The input file must be:

- In FLAC, MP3, MP4, or WAV file format

- Less than 2 hours in length

You must specify the language and format of the input file.

For best results:

- Use a lossless format, such as FLAC or WAV, with PCM 16-bit encoding.
- Use a sample rate of 8000 Hz for telephone audio.

Identifying Speakers

You can have Amazon Transcribe identify the different speakers in an audio clip, a process known as *diarization* or *speaker identification*. When you enable speaker identification, Amazon Transcribe labels each fragment with the speaker that it identified.

You can specify that Amazon Transcribe identify between 2 and 10 speakers in the audio clip. You get the best performance when the number of speakers that you ask to identify matches the number of speakers in the input audio.

To turn on speaker identification, set the `MaxSpeakerLabels` and `ShowSpeakerLabels` field of the `Settings` field when you make a call to the [StartTranscriptionJob](#) (p. 67) operation. You must set both fields or else Amazon Transcribe will return an exception.

Transcribing Streaming Audio

Streaming transcription takes a stream of your audio data and transcribes it in real time. It uses HTTP/2 streams so that the results of the transcription can be returned to your application while you send more audio to Amazon Transcribe. You can use streaming transcription when you want to make the results of live audio transcription available immediately, or when you have an audio file that you want to process as it is transcribed.

Channel Identification

When an audio file has multiple channels that you want to transcribe into separate transcriptions that identify the channel that contains the speech, use *channel identification*. For example, if you have a customer support representative on one channel and a customer on another, use channel identification to create a transcription that is identified by each channel and a single transcription that combines them.

Amazon Transcribe splits your audio file into multiple channels and transcribes the channels separately. After transcribing all channels, Amazon Transcribe also merges the transcriptions to create a single transcription. It returns all of the transcriptions in a single result file.

Speakers' utterances are ordered by their start time. An *utterance* is a unit of speech on the audio channel that is typically separated from other utterances by silence. If an utterance on one channel overlaps one on another channel, Amazon Transcribe orders them in the transcription by their start times. Utterances that overlap in the input audio don't overlap in the transcription output.

You can enable channel identification in the Amazon Transcribe console or with the API. In the console, choose **Channel identification** when you create the transcription job. When you use the API, set the `ChannelIdentification` flag when you call the [StartTranscriptionJob](#) (p. 67) operation.

The following is the abbreviated output for a conversation on two channels:

```
{
```



```

"jobName": "job id",
"accountId": "account id",
"results": {
  "transcripts": [
    {
      "transcript": "When you try ... It seems to ..."
    }
  ],
  "channel_labels": {
    "channels": [
      {
        "channel_label": "ch_0",
        "items": [
          {
            "start_time": "12.282",
            "end_time": "12.592",
            "alternatives": [
              {
                "confidence": "1.0000",
                "content": "When"
              }
            ],
            "type": "pronunciation"
          },
          {
            "start_time": "12.592",
            "end_time": "12.692",
            "alternatives": [
              {
                "confidence": "0.8787",
                "content": "you"
              }
            ],
            "type": "pronunciation"
          },
          {
            "start_time": "12.702",
            "end_time": "13.252",
            "alternatives": [
              {
                "confidence": "0.8318",
                "content": "try"
              }
            ],
            "type": "pronunciation"
          },
          Transcription abbreviated
        ]
      },
      {
        "channel_label": "ch_1",
        "items": [
          {
            "start_time": "12.379",
            "end_time": "12.589",
            "alternatives": [
              {
                "confidence": "0.5645",
                "content": "It"
              }
            ],
            "type": "pronunciation"
          },
          {
            "start_time": "12.599",
            "end_time": "12.659",

```

```

        "alternatives": [
          {
            "confidence": "0.2907",
            "content": "seems"
          }
        ],
        "type": "pronunciation"
      },
      {
        "start_time": "12.669",
        "end_time": "13.029",
        "alternatives": [
          {
            "confidence": "0.2497",
            "content": "to"
          }
        ],
        "type": "pronunciation"
      },
      Transcription abbreviated
    ]
  }
}

```

Custom Vocabularies

You can give Amazon Transcribe more information about how to process speech in your input file by creating a custom vocabulary. A *custom vocabulary* is a list of specific words that you want Amazon Transcribe to recognize in your audio input. These are generally domain-specific words and phrases, words that Amazon Transcribe isn't recognizing, or non-English names.

You specify the custom vocabulary as a list. Each entry can be a single word or a phrase. You separate the words of a phrase with a hyphen (-). For example, you type **Los Angeles** as **Los-Angeles**.

Enter acronyms or other words whose letters should be pronounced individually as single letters followed by dots, such **A.B.C.** or **C.N.N.**. To enter the plural form of an acronym, such as "ABCs", separate the "s" from the acronym with a hyphen: **"A.B.C.-s"**. You can use either upper- or lower-case letters to enter an acronym.

For more information about creating a custom vocabulary list, see [Create a Custom Vocabulary \(p. 6\)](#).

To create a custom vocabulary, use the [CreateVocabulary \(p. 49\)](#) or the [Amazon Transcribe console](#). After you submit the `CreateVocabulary` request, Amazon Transcribe processes the vocabulary. To see the processing status of the vocabulary, use the [GetVocabulary \(p. 58\)](#) operation.

To use the custom vocabulary, set the `VocabularyName` field of the `Settings` field when you make a call to the [StartTranscriptionJob \(p. 67\)](#) operation.

Create a Custom Vocabulary

A custom vocabulary is a list of words that you want Amazon Transcribe to recognize. Each entry in the list is a single word or phrase. Each entry must contain:

- fewer than 256 characters
- only characters from the allowed character set

For valid character sets, see [English Character Set \(p. 7\)](#), [French Character Set \(p. 7\)](#), [Italian Character Set \(p. 8\)](#), [Portuguese Character Set \(p. 9\)](#), and [Spanish Character Set \(p. 10\)](#).

The size limit for a custom vocabulary is 50 KB.

Create a custom vocabulary by using the [CreateVocabulary \(p. 49\)](#) operation or the [Amazon Transcribe console](#). If you use the Amazon Transcribe console to create a custom vocabulary, you can provide the entries in either a text file or a comma-separated values file (CSV).

Creating a Custom Vocabulary with a Text File (Console)

When you use a text file, place each word or phrase on a separate line. Save the file with the extension .txt. The following is an example input file in text format:

```
apple
bear
coffee-dog
five
earring
good-morning
hi
Etienne
A.B.C.
A.B.C.-s
```

Creating a Custom Vocabulary with a Comma-separated Values File (Console)

In a comma-separated values (CSV) file, separate each word or phrase with a comma. You can put multiple entries on one line, and use line returns to break long lines. Save the file with the extension ".csv".

The following is an example input file in CSV format:

```
apple,bear,coffee-dog,
five,earring,good-morning,
hi,Etienne,A.B.C.,A.B.C.-s
```

English Character Set

For English custom vocabularies, you can use the following characters:

- a - z
- A - Z
- - (hyphen)

French Character Set

For French custom vocabularies, you can use the following characters:

- a - z
- A - Z
- ' (apostrophe)

- - (hyphen)

Character	Code	Character	Code
À	00C0	à	00E0
Â	00C2	â	00E2
Ç	00C7	ç	00E7
È	00C8	è	00E8
É	00C9	é	00E9
Ê	00CA	ê	00EA
Ë	00CB	ë	00EB
Î	00CE	î	00EE
Ï	00CF	ï	00EF
Ô	00D4	ô	00F4
Ö	00D6	ö	00F6
Ù	00D9	ù	00F9
Û	00DB	û	00FB
Ü	00DC	ü	00FC

Italian Character Set

For Italian custom vocabularies, you can use the following characters:

- a - z
- A - Z
- ' (apostrophe)
- - (hyphen)

You can also use the following Unicode characters:

Character	Code	Character	Code
À	00C0	à	00E0
Ä	00C4	ä	00E4
Ç	00C7	ç	00E7
È	00C8	è	00E8
É	00C9	é	00E9
Ê	00CA	ê	00EA

Character	Code	Character	Code
Ë	00CB	ë	00EB
Ì	00CC	ì	00EC
Ò	00D2	ò	00F2
Ù	00D9	ù	00F9
Ü	00DC	ü	00FC

Portuguese Character Set

For Portuguese custom vocabularies, you can use the following characters:

- a - z
- A - Z
- ' (apostrophe)
- - (hyphen)

You can also use the following Unicode characters:

Character	Code	Character	Code
À	00C0	à	00E0
Á	00C1	á	00E1
Â	00C2	â	00E2
Ã	00C3	ã	00E3
Ä	00C4	ä	00E4
Ç	00C7	ç	00E7
È	00C8	è	00E8
É	00C9	é	00E9
Ê	00CA	ê	00EA
Ë	00CB	ë	00EB
Í	00CD	í	00ED
Ñ	00D1	ñ	00F1
Ó	00D3	ó	00F3
Ô	00D4	ô	00F4
Õ	00D5	õ	00F5
Ö	00D6	ö	00F6
Ú	00DA	ú	00FA

Character	Code	Character	Code
Ü	00DC	ü	00FC

Spanish Character Set

For Spanish custom vocabularies, you can use the following characters:

- a - z
- A - Z
- ' (apostrophe)
- - (hyphen)

You can also use the following Unicode characters:

Character	Code	Character	Code
Á	00C1	á	00E1
É	00C9	é	00E9
Í	00CD	ë	00ED
Ó	00D3	ó	0XF3
Ú	00DA	ú	00FA
Ñ	00D1	ñ	0XF1
ü	00FC		

Getting Started with Amazon Transcribe

To get started using Amazon Transcribe, set up an AWS account and create an AWS Identity and Access Management (IAM) user. To use the AWS Command Line Interface (AWS CLI), download and configure it.

Topics

- [Step 1: Set up an AWS Account and Create an Administrator User \(p. 11\)](#)
- [Step 2: Set up the AWS Command Line Interface \(AWS CLI\) \(p. 12\)](#)
- [Step 3: Getting Started Using the Console \(p. 13\)](#)
- [Step 4: Getting Started Using the API \(p. 17\)](#)
- [Step 5: Getting Started With Streaming Audio \(p. 21\)](#)

Step 1: Set up an AWS Account and Create an Administrator User

Before you use Amazon Transcribe for the first time, complete the following tasks:

1. [Sign up for AWS \(p. 11\)](#)
2. [Create an IAM User \(p. 12\)](#)

Sign up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all AWS services, including Amazon Transcribe. You are charged only for the services that you use.

With Amazon Transcribe, you pay only for the resources that you use. If you are a new AWS customer, you can get started with Amazon Transcribe for free. For more information, see [AWS Free Usage Tier](#).

If you already have an AWS account, skip to the next section.

To create an AWS account

1. Open <https://aws.amazon.com/>, and then choose **Create an AWS Account**.

Note

If you previously signed in to the AWS Management Console using AWS account root user credentials, choose **Sign in to a different account**. If you previously signed in to the console using IAM credentials, choose **Sign-in using root account credentials**. Then choose **Create a new AWS account**.

2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code using the phone keypad.

Record your AWS account ID because you'll need it for the next task.

Create an IAM User

Services in AWS, such as Amazon Transcribe, require that you provide credentials when you access them. This allows the service to determine whether you have permissions to access the service's resources.

We strongly recommend that you access AWS using AWS Identity and Access Management (IAM), not the credentials for your AWS account. To use IAM to access AWS, create an IAM user, add the user to an IAM group with administrative permissions, and then grant administrative permissions to the IAM user. You can then access AWS using a special URL and the IAM user's credentials.

The Getting Started exercises in this guide assume that you have a user with administrator privileges, `adminuser`.

To create an administrator user and sign in to the console

1. Create an administrator user called `adminuser` in your AWS account. For instructions, see [Creating Your First IAM User and Administrators Group](#) in the *IAM User Guide*.
2. Sign in to the AWS Management Console using a special URL. For more information, see [How Users Sign In to Your Account](#) in the *IAM User Guide*.

For more information about IAM, see the following:

- [AWS Identity and Access Management \(IAM\)](#)
- [Getting Started](#)
- [IAM User Guide](#)

Next Step

[Step 2: Set up the AWS Command Line Interface \(AWS CLI\) \(p. 12\)](#)

Step 2: Set up the AWS Command Line Interface (AWS CLI)

You don't need the AWS CLI to perform the steps in the Getting Started exercises. However, some of the other exercises in this guide do require it. If you prefer, you can skip this step and set up the AWS CLI later.

To set up the AWS CLI

1. Download and configure the AWS CLI. For instructions, see the following topics in the *AWS Command Line Interface User Guide*:
 - [Getting Set Up with the AWS Command Line Interface](#)
 - [Configuring the AWS Command Line Interface](#)
2. In the AWS CLI `config` file, add a named profile for the administrator user:

```
[profile adminuser]
aws_access_key_id = adminuser access key ID
aws_secret_access_key = adminuser secret access key
region = aws-region
```


You use this profile when executing the AWS CLI commands. For more information about named profiles, see [Named Profiles](#) in the *AWS Command Line Interface User Guide*. For a list of AWS Regions, see [Regions and Endpoints](#) in the *Amazon Web Services General Reference*.

3. Verify the setup by typing the following help command at the command prompt:

```
aws help
```

Next Step

[Step 3: Getting Started Using the Console \(p. 13\)](#)

Step 3: Getting Started Using the Console

The easiest way to get started with Amazon Transcribe is to submit a job using the console to transcribe an audio file. If you haven't reviewed the concepts and terminology in [How Amazon Transcribe Works \(p. 3\)](#), we recommend that you do that before proceeding.

Topics

- [Create a Transcription Job \(p. 13\)](#)
- [View a Transcription Job \(p. 14\)](#)

Create a Transcription Job

Use the Amazon Transcribe console to create a transcription job for your audio files.

1. Provide the following information:
 - **Transcription job name**—A name for the job. The name must be unique within your AWS account.
 - **Amazon S3 input URL**—The Amazon S3 location of your input audio file. The location must be in the same region as the endpoint that you are calling.
 - **Language**—Choose the language of your input file. Amazon Transcribe can transcribe English and Spanish audio files.
 - **Format**—The format of the audio file. For best results you should use a lossless format such as FLAC or WAV with PCM 16-bit encoding.
 - **Media sampling rate (Hz)**—Optional. The bit sampling rate of the audio file. Amazon Transcribe accepts sample rates between 8000 Hz and 48000 Hz. For best results, you should use 8000 Hz for low-fidelity audio and 16000 for high-fidelity audio.

The following shows the **Create Transcription Job** filled out for a sample job.

Create transcription job

Input [Info](#)

Name

The name can be up to 200 characters long. Valid characters are a-z, A-Z, 0-9 and - (hyphen).

S3 input URL
Type or paste the URL of your input audio file in S3.

Valid formats for audio files are mp3, mp4, wav, and flac.

Language
Choose the language of the input audio.

English ▼

Format
Choose the format of your audio file.

mp3 ▼

Valid formats for the audio are mp3, mp4, wav and flac.

Audio sampling rate (Hz)
Type the sampling rate of the input audio file.

Must be an integer between 8000 and 48000

Apply custom vocabulary [Info](#)
A custom vocabulary improves the accuracy of recognizing words, phrases, and commands.

▼

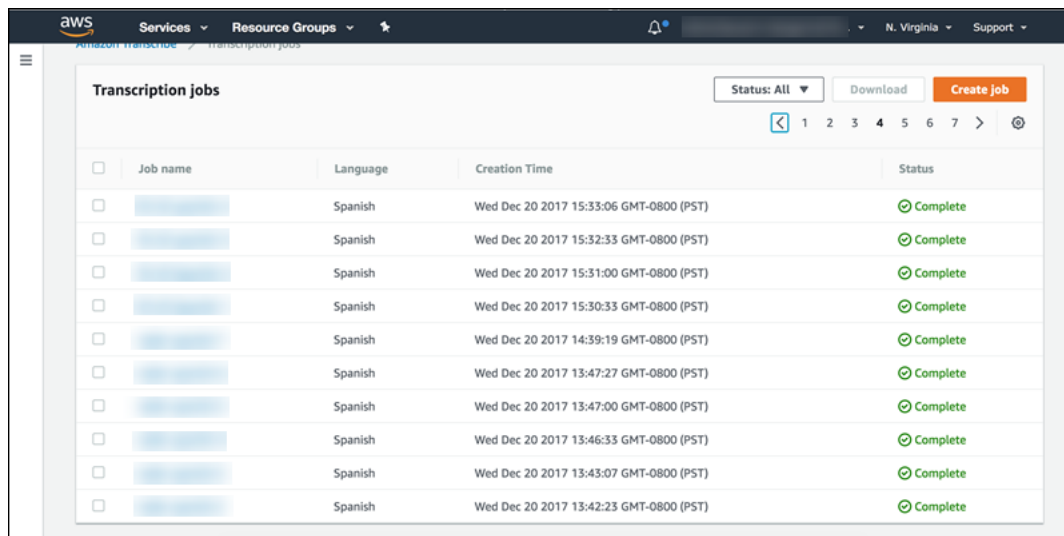
Speaker identification [Info](#)
Identifies speakers in the input audio file.

☒ Disabled
☐ Enabled

2. Choose **Create** to submit the job for processing.

View a Transcription Job

Completed transcription jobs are displayed in a list that contains a brief description of the job. The **Availability** column shows the remaining time that the job results will be kept on the server. Jobs are kept for 90 days and then deleted from the system.



The screenshot shows the Amazon Transcribe console interface. At the top, there's a navigation bar with the AWS logo, 'Services', 'Resource Groups', and a search bar. Below this, the 'Transcription jobs' page is displayed. It features a table with columns for 'Job name', 'Language', 'Creation Time', and 'Status'. There are 10 jobs listed, all with a status of 'Complete'. The table is paginated, showing jobs 1 through 7. To the right of the table, there are buttons for 'Status: All', 'Download', and 'Create job'.

Job name	Language	Creation Time	Status
[Redacted]	Spanish	Wed Dec 20 2017 15:33:06 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 15:32:33 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 15:31:00 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 15:30:33 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 14:39:19 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 13:47:27 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 13:47:00 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 13:46:33 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 13:43:07 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 13:42:23 GMT-0800 (PST)	Complete

Choose a job in the list to see information about the job.

The information page about the transcription job has three sections. The **Detail** section provides details about the transcription job, including the name, information about when the job will be deleted from the server, and the input and output URLs. Use the output URL to download the output from your transcription job.

The **Output** section contains the transcription of the audio submitted to Amazon Transcribe. You can download the transcription by choosing the **Download transcription** button.



The screenshot shows the 'Details' page for a specific transcription job named 'ES-US-spanish-4'. The page has a breadcrumb trail: 'Amazon Transcribe > Transcription jobs > ES-US-spanish-4'. The 'Details' section includes the following information:

- Job name:** [Redacted]
- Status:** Complete (indicated by a green checkmark icon)
- Expiration date:** The transcription will be available for another 9 days
- Input URL:** [https://s3.amazonaws.com/\[Redacted\].wav](https://s3.amazonaws.com/[Redacted].wav)
- Output URL:** [https://s3.amazonaws.com/\[Redacted\]/asrOutput.json](https://s3.amazonaws.com/[Redacted]/asrOutput.json)

A 'Download transcription' button is located in the top right corner of the details section.

The **Code samples** section contains the JSON input for the [StartTranscriptionJob](#) (p. 67) operation and the output from the [GetTranscriptionJob](#) (p. 56) operation.

Code Samples

Audio conversion

JSON Request

```
{
  "TranscriptionJobName": "██████████",
  "LanguageCode": "en-US",
  "MediaSampleRateHertz": 8000,
  "MediaFormat": "wav",
  "Media": {
    "MediaFileUri": "https://██████████.s3.amazonaws.com/answer2.wav"
  }
}
```

JSON Response

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "[REDACTED]",
    "TranscriptionJobStatus": "COMPLETED",
    "LanguageCode": "en-US",
    "MediaSampleRateHertz": 8000,
    "MediaFormat": "wav",
    "Media": {
      "MediaFileUri": "https://[REDACTED]
[REDACTED]/answer2.wav?X-Amz-Security-
Token=FQoDYXdzEOD%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaDJVU%2Fub%2B5SJstTh3YXiK3AzA%2FWgl"
```

Next Step

Step 4: Getting Started Using the API (p. 17)

Step 4: Getting Started Using the API

This section contains examples that demonstrate using the Amazon Transcribe API. You can use these samples to learn about the API or as building blocks in your own applications.

Topics

- [Getting Started \(AWS Command Line Interface\) \(p. 17\)](#)
- [Getting Started \(AWS SDK for Python \(Boto\)\) \(p. 19\)](#)

Getting Started (AWS Command Line Interface)

In the following exercise, you use the AWS Command Line Interface (AWS CLI) to transcribe speech into text. To complete this exercise, you need to:

- Have a text editor.
- Be familiar with the AWS CLI. For more information, see [Step 2: Set up the AWS Command Line Interface \(AWS CLI\) \(p. 12\)](#).
- Have a speech file in .WAV or .MP4 format that is stored in an S3 bucket that has the proper permissions. For more information about the permissions needed for Amazon Transcribe, see [Using Identity-based Policies \(IAM Policies\) for Amazon Transcribe \(p. 42\)](#).

To transcribe text, you have to provide the input parameters in a JSON file.

To transcribe text

1. Copy your input speech to an S3 bucket. The location must be in the same region as the endpoint that you are calling. This example assumes that the file is in an S3 bucket named `test-transcribe` and that the file name is `answer2.wav`.
2. Create a JSON file named `test-start-command.json` that contains the input parameters for the [StartTranscriptionJob \(p. 67\)](#) operation.

```
{
  "TranscriptionJobName": "request ID",
  "LanguageCode": "en-US",
  "MediaFormat": "wav",
  "Media": {
    "MediaFileUri": "https://S3 endpoint/test-transcribe/answer2.wav"
  }
}
```

3. In the AWS CLI, run the following command. The example is formatted for Unix, Linux, and macOS. For Windows, replace the backslash (\) Unix continuation character at the end of each line with a caret (^).

```
aws transcribe start-transcription-job \
  --region region \
  --cli-input-json file://test-start-command.json
```

Amazon Transcribe responds with the following:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "request ID",
    "LanguageCode": "en-US",
```

```
    "TranscriptionJobStatus": "IN_PROGRESS",
    "Media": {
      "MediaFileUri": "https://S3 endpoint/test-transcribe/answer2.wav"
    },
    "CreationTime": timestamp,
    "MediaFormat": "wav"
  }
}
```

To list transcription jobs

- Run the following command:

```
aws transcribe list-transcription-jobs \
  --region region \
  --status IN_PROGRESS
```

Amazon Transcribe responds with the following:

```
{
  "Status": "IN_PROGRESS",
  "TranscriptionJobSummaries": [
    {
      "TranscriptionJobName": "request ID",
      "LanguageCode": "en-US",
      "CreationTime": timestamp,
      "TranscriptionJobStatus": "IN_PROGRESS"
    }
  ]
}
```

To get the results of a transcription job

- When the job has the status COMPLETED, get the results of the job. Type the following command:

```
aws transcribe get-transcription-job \
  --region region \
  --transcription-job-name "request ID"
```

Amazon Transcribe responds with the following:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "request ID",
    "LanguageCode": "en-US",
    "TranscriptionJobStatus": "COMPLETED",
    "Media": {
      "MediaFileUri": "input URI"
    },
    "CreationTime": timestamp,
    "CompletionTime": timestamp,
    "Transcript": {
      "TranscriptFileUri": "output URI"
    }
  }
}
```

2. Use the output URI to get the transcribed text from the audio file. The following is the output from transcribing a short audio clip:

```
{
  "jobName": "job ID",
  "accountId": "account ID",
  "results": {
    "transcripts": [
      {
        "transcript": " that's no answer"
      }
    ],
    "items": [
      {
        "start_time": "0.180",
        "end_time": "0.470",
        "alternatives": [
          {
            "confidence": 0.84,
            "content": "that's"
          }
        ],
        "type": "pronunciation"
      },
      {
        "start_time": "0.470",
        "end_time": "0.710",
        "alternatives": [
          {
            "confidence": 0.99,
            "content": "no"
          }
        ],
        "type": "pronunciation"
      },
      {
        "start_time": "0.710",
        "end_time": "1.080",
        "alternatives": [
          {
            "confidence": 0.874,
            "content": "answer"
          }
        ],
        "type": "pronunciation"
      }
    ]
  },
  "status": "COMPLETED"
}
```

Getting Started (AWS SDK for Python (Boto))

In this exercise you create script that uses the SDK for Python to transcribe speech into text. To complete this exercise, you need to:

- Install the AWS CLI. For more information, see [Step 2: Set up the AWS Command Line Interface \(AWS CLI\) \(p. 12\)](#). This installs the AWS SDK for Python (Boto).
- Have a speech file in .WAV or .MP4 format that is stored in an S3 bucket that has the proper permissions. For more information about the permissions needed for Amazon Transcribe, see [Using Identity-based Policies \(IAM Policies\) for Amazon Transcribe \(p. 42\)](#). The location must be in the

same region as the endpoint that you are calling. This example assumes that the file is in an Amazon S3 bucket named `test-transcribe` and that the file name is `answer2.wav`.

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe')
job_name = "job name"
job_uri = "https://S3 endpoint/test-transcribe/answer2.wav"
transcribe.start_transcription_job(
    TranscriptionJobName=job_name,
    Media={'MediaFileUri': job_uri},
    MediaFormat='wav',
    LanguageCode='en-US'
)
while True:
    status = transcribe.get_transcription_job(TranscriptionJobName=job_name)
    if status['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

When the transcription job is complete, the result links to an Amazon S3 presigned URL that contains the transcription in JSON format:

```
{
  "jobName": "job ID",
  "accountId": "account ID",
  "results": {
    "transcripts": [
      {
        "transcript": " that's no answer",
        "confidence": 1.0
      }
    ],
    "items": [
      {
        "start_time": "0.180",
        "end_time": "0.470",
        "alternatives": [
          {
            "confidence": 0.84,
            "word": "that's"
          }
        ]
      },
      {
        "start_time": "0.470",
        "end_time": "0.710",
        "alternatives": [
          {
            "confidence": 0.99,
            "word": "no"
          }
        ]
      },
      {
        "start_time": "0.710",
        "end_time": "1.080",
        "alternatives": [
          {
            "confidence": 0.87,
```



```
        "word": "answer"
      }
    ]
  },
  "status": "COMPLETED"
}
```

Step 5: Getting Started With Streaming Audio

The following example is a Java program that transcribes streaming audio. The input comes from your computer's microphone or a file upload and the output is presented on your computer's standard output.

To run this example, you need the following:

- You must use the [AWS SDK for Java 2.x](#)
- Clients must use Java 1.8 to be compatible with the AWS SDK for Java 2.x.

```
/**
 * COPYRIGHT:
 *
 * Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
package com.amazonaws.transcribestreaming;

public class TranscribeStreamingDemoApp {
    private static final String REGION = "region";

    private static TranscribeStreamingAsyncClient client;

    public static void main(String args[]) throws URISyntaxException, ExecutionException,
        InterruptedException, LineUnavailableException {

        client = TranscribeStreamingAsyncClient.builder()
            .credentialsProvider(getCredentials())
            .region(REGION)
            .build();

        CompletableFuture<Void> result =
            client.startStreamTranscription(getRequest(16_000),
                new AudioStreamPublisher(getStreamFromMic()),
                getResponseHandler());

        result.get();
        client.close();
    }

    private static InputStream getStreamFromMic() throws LineUnavailableException {
```

```
// Signed PCM AudioFormat with 16kHz, 16 bit sample size, mono
int sampleRate = 16000;
AudioFormat format = new AudioFormat(sampleRate, 16, 1, true, false);
DataLine.Info info = new DataLine.Info(TargetDataLine.class, format);

if (!AudioSystem.isLineSupported(info)) {
    System.out.println("Line not supported");
    System.exit(0);
}

TargetDataLine line = (TargetDataLine) AudioSystem.getLine(info);
line.open(format);
line.start();

InputStream audioStream = new AudioInputStream(line);
return audioStream;
}

private InputStream getStreamFromFile(String audioFileName) {
    try {
        File inputFile = new
File(getClass().getClassLoader().getResource(audioFileName).getFile());
        InputStream audioStream = new FileInputStream(inputFile);
        return audioStream;
    } catch (FileNotFoundException e) {
        throw new RuntimeException(e);
    }
}

private static AwsCredentialsProvider getCredentials() {
    return DefaultCredentialsProvider.create();
}

private static StartStreamTranscriptionRequest getRequest(Integer mediaSampleRateHertz)
{
    return StartStreamTranscriptionRequest.builder()
        .languageCode(LanguageCode.EN_US.toString())
        .mediaEncoding(MediaEncoding.PCM)
        .mediaSampleRateHertz(mediaSampleRateHertz)
        .build();
}

private static StartStreamTranscriptionResponseHandler getResponseHandler() {
    return StartStreamTranscriptionResponseHandler.builder()
        .onResponse(r -> {
            System.out.println("Received Initial response");
        })
        .onError(e -> {
            System.out.println(e.getMessage());
            System.out.println("Error Occurred: " +
ExceptionUtils.getStackTrace(e.fillInStackTrace()));
        })
        .onComplete(() -> {
            System.out.println("=== All records stream successfully ===");
        })
        .subscriber(event -> {
            List<Result> results = ((TranscriptEvent)
event).transcript().results();
            if(results.size()>0) {
                if(!results.get(0).alternatives().get(0).transcript().isEmpty()) {
System.out.println(results.get(0).alternatives().get(0).transcript());
                }
            }
        })
}
```

```
        .build();
    }

    private static class AudioStreamPublisher implements Publisher<AudioStream> {
        private final InputStream inputStream;

        private AudioStreamPublisher(InputStream inputStream) {
            this.inputStream = inputStream;
        }

        @Override
        public void subscribe(Subscriber<? super AudioStream> s) {
            if (currentSubscription == null) {
                this.currentSubscription = new SubscriptionImpl(s, inputStream);
            } else {
                this.currentSubscription.cancel();
                this.currentSubscription = new SubscriptionImpl(s, inputStream);
            }
            s.onSubscribe(currentSubscription);
        }
    }

    private static class SubscriptionImpl implements Subscription {
        private static final int CHUNK_SIZE_IN_BYTES = 1024 * 1;
        private ExecutorService executor = Executors.newFixedThreadPool(1);
        private AtomicLong demand = new AtomicLong(0);

        private final Subscriber<? super AudioStream> subscriber;
        private final InputStream inputStream;

        private SubscriptionImpl(Subscriber<? super AudioStream> s, InputStream
inputStream) {
            this.subscriber = s;
            this.inputStream = inputStream;
        }

        @Override
        public void request(long n) {
            if (n <= 0) {
                subscriber.onError(new IllegalArgumentException("Demand must be
positive"));
            }

            demand.getAndAdd(n);

            executor.submit(() -> {
                try {
                    do {
                        ByteBuffer audioBuffer = getNextEvent();
                        if (audioBuffer.remaining() > 0) {
                            AudioEvent audioEvent = audioEventFromBuffer(audioBuffer);
                            subscriber.onNext(audioEvent);
                        } else {
                            subscriber.onComplete();
                            break;
                        }
                    } while (demand.decrementAndGet() > 0);
                } catch (Exception e) {
                    subscriber.onError(e);
                }
            });
        }

        @Override
        public void cancel() {
            executor.shutdown();
        }
    }
}
```

```
    }

    private ByteBuffer getNextEvent() {
        ByteBuffer audioBuffer = null;
        byte[] audioBytes = new byte[CHUNK_SIZE_IN_BYTES];

        int len = 0;
        try {
            len = inputStream.read(audioBytes);

            if (len <= 0) {
                audioBuffer = ByteBuffer.allocate(0);
            } else {
                audioBuffer = ByteBuffer.wrap(audioBytes, 0, len);
            }
        } catch (IOException e) {
            throw new UncheckedIOException(e);
        }

        return audioBuffer;
    }

    private AudioEvent audioEventFromBuffer(ByteBuffer bb) {
        return AudioEvent.builder()
            .audioChunk(SdkBytes.fromByteBuffer(bb))
            .build();
    }
}
```

Streaming Transcription

Amazon Transcribe streaming transcription enables you to send an audio stream and receive a stream of text in real-time. The API makes it easy for developers to add real-time speech-to-text capability to their applications.

Amazon Transcribe streaming transcription can be used for a variety of purposes. For example:

- Streaming transcriptions can generate real-time subtitles for live broadcast media.
- Lawyers can make real-time annotations on top of streaming transcriptions during courtroom depositions.
- Video game chat can be transcribed in real time so that hosts can moderate content or run real-time analysis.
- Streaming transcriptions can provide assistance to the hearing-impaired.

To make it easier to get started, we provide a streaming client that handles retrying the connection when there are transient problems on the network. You can use this client as a starting point for your own applications.

Topics

- [Using Amazon Transcribe Streaming \(p. 25\)](#)
- [A Streaming Retry Client \(p. 27\)](#)
- [Using the Retry Client \(p. 32\)](#)

Using Amazon Transcribe Streaming

Streaming transcription takes a stream of your audio data and transcribes it in real time. The transcription is returned to your application in a stream of transcription events.

To start transcribing streaming audio, use the [StartStreamTranscription](#) operation. You can transcribe up to four hours of streaming audio.

Amazon Transcribe breaks your incoming audio stream based on natural speech segments, such as a change in speaker or a pause in the audio. The transcription is returned progressively to your application, with each response containing more transcribed speech until the entire segment is transcribed.

In the following example, each line is a partial result transcription output of an audio segment being streamed:

```
the amazon is the largest
the amazon is the largest
the amazon is the largest
the amazon is the largest rainforest
the amazon is the largest rainforest
the amazon is the largest rainforest
the amazon is the largest rainforest on the
the amazon is the largest rainforest on the
the amazon is the largest rainforest on the planet
the amazon is the largest rainforest on the planet
the amazon is the largest rainforest on the planet
the amazon is the largest rainforest on the planet
the amazon is the largest rainforest on the planet covering over
the amazon is the largest rainforest on the planet covering over
```

```
the amazon is the largest rainforest on the planet covering over two million
```

Each [Result](#) object in the response contains a field called `IsPartial` that indicates whether the response is a partial response containing the transcription results so far or if it is a complete transcription of the audio segment.

Each [Result](#) object also contains the start time and end time of the term from the audio stream so that you can, for example, synchronize the transcription with the video.

The following example is a partial transcription response from the [StartStreamTranscription](#) operation.

```
{
  "TranscriptResultStream": {
    "TranscriptEvent": {
      "Transcript": {
        "Results": [
          {
            "Alternatives": [
              {
                "Items": [
                  {
                    "Content": "the",
                    "EndTime": 0.3799375,
                    "StartTime": 0.0299375,
                    "Type": "pronunciation"
                  },
                  {
                    "Content": "amazon",
                    "EndTime": 0.5899375,
                    "StartTime": 0.3899375,
                    "Type": "pronunciation"
                  },
                  {
                    "Content": "is",
                    "EndTime": 0.7899375,
                    "StartTime": 0.5999375,
                    "Type": "pronunciation"
                  },
                  {
                    "Content": "the",
                    "EndTime": 0.9199375,
                    "StartTime": 0.7999375,
                    "Type": "pronunciation"
                  },
                  {
                    "Content": "largest",
                    "EndTime": 1.0199375,
                    "StartTime": 0.9299375,
                    "Type": "pronunciation"
                  }
                ],
                "Transcript": "the amazon is the largest"
              }
            ],
            "EndTime": 1.02,
            "IsPartial": true,
            "ResultId": "2db76dc8-d728-11e8-9f8b-f2801f1b9fd1",
            "StartTime": 0.0199375
          }
        ]
      }
    }
  }
}
```

A Streaming Retry Client

You can use the following code in your applications to handle retry logic for Amazon Transcribe streaming transcription. The code provides tolerance for intermittent failures in the connection to Amazon Transcribe. There are two parts of the client: an interface that you implement for your application, and the retry client itself.

Streaming Retry Client Code

This code implements a streaming retry client. It manages the connection to Amazon Transcribe and retries sending data when there are errors on the connection. For example, if there is a transient error on the network this client will resend the request that failed.

The retry client has two properties that control the behavior of the client. You can set:

- The maximum number of times that the client should attempt before failing. Reduce this value to make your application stop retrying sooner when there are network issues. The default is 10.
- The time in milliseconds that the client should wait between retries. Longer times raise the risk of losing data, shorter times raise the risk of your application being throttled. The default is 100 milliseconds.

The following is the client. You can copy this code to your application or use it as a starting point for your own client.

```
/**
 * COPYRIGHT:
 *
 * Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
package com.amazonaws.transcribe.streaming.retryclient;

/**
 * Build a client wrapper around the Amazon Transcribe client to retry
 * on an exception that can be retried.
 */
public class TranscribeStreamingRetryClient {

    private static final int DEFAULT_MAX_RETRIES = 10;
    private static final int DEFAULT_MAX_SLEEP_TIME_MILLS = 100;
    private int maxRetries = DEFAULT_MAX_RETRIES;
    private int sleepTime = DEFAULT_MAX_SLEEP_TIME_MILLS;
    private final TranscribeStreamingAsyncClient client;
    List<Class<?>> nonRetriableExceptions = Arrays.asList(BadRequestException.class);

    private static final Logger log =
    LoggerFactory.getLogger(TranscribeStreamingRetryClient.class);
    /**
     * Create a TranscribeStreamingRetryClient with given credential and configuration
     * @param creds Creds to use for transcription
     */
}
```

```
* @param endpoint Endpoint to use for transcription
* @param region Region to use for transcriptions
* @throws URISyntaxException if the endpoint is not a URI
*/
public TranscribeStreamingRetryClient(AwsCredentialsProvider creds,
                                     String endpoint, Region region) throws
URISyntaxException {
    this(TranscribeStreamingAsyncClient.builder()
        .overrideConfiguration(
            c -> c.putAdvancedOption(
                SdkAdvancedClientOption.SIGNER,
                EventStreamAws4Signer.create()))
        .credentialsProvider(creds)
        .endpointOverride(new URI(endpoint))
        .region(region)
        .build());
}

/**
 * Initiate TranscribeStreamingRetryClient with TranscribeStreamingAsyncClient
 * @param client TranscribeStreamingAsyncClient
 */
public TranscribeStreamingRetryClient(TranscribeStreamingAsyncClient client) {
    this.client = client;
}

/**
 * Get Max retries
 * @return Max retries
 */
public int getMaxRetries() {
    return maxRetries;
}

/**
 * Set Max retries
 * @param maxRetries Max retries
 */
public void setMaxRetries(int maxRetries) {
    this.maxRetries = maxRetries;
}

/**
 * Get sleep time
 * @return sleep time between retries
 */
public int getSleepTime() {
    return sleepTime;
}

/**
 * Set sleep time between retries
 * @param sleepTime sleep time
 */
public void setSleepTime(int sleepTime) {
    this.sleepTime = sleepTime;
}

/**
 * Initiate a Stream Transcription with retry.
 * @param request StartStreamTranscriptionRequest to use to start transcription
 * @param publisher The source audio stream as Publisher
 * @param responseHandler StreamTranscriptionBehavior object that defines how the
response needs to be handled.
 * @return Completable future to handle stream response.
 */
```



```
    public CompletableFuture<Void> startStreamTranscription(final
StartStreamTranscriptionRequest request,
                                                                    final Publisher<AudioStream>
publisher,
                                                                    final
StreamTranscriptionBehavior responseHandler) {

        CompletableFuture<Void> finalFuture = new CompletableFuture<>();

        recursiveStartStream(rebuildRequestWithSession(request), publisher,
responseHandler, finalFuture, 0);

        return finalFuture;
    }

    /**
     * Recursively call startStreamTranscription() to be called till the request is
     completed or till we run out of retries.
     * @param request StartStreamTranscriptionRequest
     * @param publisher The source audio stream as Publisher
     * @param responseHandler StreamTranscriptionBehavior object that defines how the
     response needs to be handled.
     * @param finalFuture final future to finish on completing the chained futures.
     * @param retryAttempt Current attempt number
     */
    private void recursiveStartStream(final StartStreamTranscriptionRequest request,
                                                                    final Publisher<AudioStream> publisher,
                                                                    final StreamTranscriptionBehavior responseHandler,
                                                                    final CompletableFuture<Void> finalFuture,
                                                                    final int retryAttempt) {

        CompletableFuture<Void> result = client.startStreamTranscription(request,
publisher,
        getResponseHandler(responseHandler));
        result.whenComplete((r, e) -> {
            if (e != null) {
                log.debug("Error occurred:", e);

                if (retryAttempt <= maxRetries && isExceptionRetriable(e)) {
                    log.debug("Retriable error occurred and will be retried.");
                    log.debug("Sleeping for sometime before retrying...");
                    try {
                        Thread.sleep(sleepTime);
                    } catch (InterruptedException e1) {
                        log.debug("Unable to sleep. Failed with exception: ", e);
                        e1.printStackTrace();
                    }
                    log.debug("Making retry attempt: " + (retryAttempt+1));
                    recursiveStartStream(request, publisher, responseHandler, finalFuture,
retryAttempt + 1);
                } else {
                    log.error("Encountered unretriable exception or ran out of retries. ");
                    responseHandler.onError(e);
                    finalFuture.completeExceptionally(e);
                }
            } else {
                responseHandler.onComplete();
                finalFuture.complete(null);
            }
        });
    }

    private StartStreamTranscriptionRequest
rebuildRequestWithSession(StartStreamTranscriptionRequest request) {
        return StartStreamTranscriptionRequest.builder()
            .languageCode(request.languageCode())
            .mediaEncoding(request.mediaEncoding())
    }
```

```
        .mediaSampleRateHertz(request.mediaSampleRateHertz())
        .sessionId(UUID.randomUUID().toString())
        .build();
    }

    /**
     * StartStreamTranscriptionResponseHandler implements subscriber of transcript stream
     * Output is printed to standard output
     */
    private StartStreamTranscriptionResponseHandler getResponseHandler(
        StreamTranscriptionBehavior transcriptionBehavior) {
        final StartStreamTranscriptionResponseHandler build =
        StartStreamTranscriptionResponseHandler.builder()
            .onResponse(r -> {
                transcriptionBehavior.onResponse(r);
            })
            .onError(e -> {
                //Do nothing here. Don't close any streams that shouldn't be cleaned up
yet.
            })
            .onComplete(() -> {
                //Do nothing here. Don't close any streams that shouldn't be cleaned up
yet.
            })
            .subscriber(event -> transcriptionBehavior.onStream(event))
            .build();
        return build;
    }

    /**
     * Check if the exception can be retried.
     * @param e Exception that occurred
     * @return True if the exception is retrieable
     */
    private boolean isExceptionRetriable(Throwable e) {
        e.printStackTrace();
        if (nonRetriableExceptions.contains(e.getClass())) {
            return false;
        }
        return true;
    }

    public void close() {
        this.client.close();
    }
}
}
```

Streaming Retry Client Interface Code

This interface is similar to the response handler used in the getting started example. It implements the same event handlers. Implement this interface to use the streaming retry client.

```
/**
 * COPYRIGHT:
 *
 * Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
```

```
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*
*/
package com.amazonaws.wolverine.streaming.retryclient;

/**
 * Defines how a stream response should be handled.
 * You should build a class implementing this interface to define the behavior.
 */
public interface StreamTranscriptionBehavior {
    /**
     * Defines how to respond when encountering an error on the stream transcription.
     * @param e The exception
     */
    void onError(Throwable e);

    /**
     * Defines how to respond to the Transcript result stream.
     * @param e The TranscriptResultStream event
     */
    void onStream(TranscriptResultStream e);

    /**
     * Defines what to do on initiating a stream connection with the service.
     * @param r StartStreamTranscriptionResponse
     */
    void onResponse(StartStreamTranscriptionResponse r);

    /**
     * Defines what to do on stream completion
     */
    void onComplete();
}
```

The following is an example implementation of the `StreamTranscriptionBehavior` interface. You can use this implementation or use it as a starting point for your own implementation.

```
/**
 * Implementation of StreamTranscriptionBehavior to define how a stream response is
 * handled.
 *
 * *
 * COPYRIGHT:
 *
 * Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
package com.amazonaws.wolverine.streaming.retryclient.example;

public class StreamTranscriptionBehaviorImpl implements StreamTranscriptionBehavior {
```

```
@Override
public void onError(Throwable e) {
    System.out.println("=== Failure encountered ===");
    e.printStackTrace();
}

@Override
public void onStream(TranscriptResultStream e) {
    // EventResultStream has other fields related to the timestamp of the transcripts
    in it.
    // Please refer to the javadoc of TranscriptResultStream for more details
    List<Result> results = ((TranscriptEvent) e).transcript().results();
    if (results.size() > 0) {
        if (results.get(0).alternatives().size() > 0)
            if (!results.get(0).alternatives().get(0).transcript().isEmpty()) {
                System.out.println(results.get(0).alternatives().get(0).transcript());
            }
    }
}

@Override
public void onResponse(StartStreamTranscriptionResponse r) {

    System.out.println(String.format("=== Received initial response. Request Id: %s
===", r.requestId()));
}

@Override
public void onComplete() {
    System.out.println("=== All records streamed successfully ===");
}
}
```

Next step

[Using the Retry Client \(p. 32\)](#)

Using the Retry Client

The following is a sample application that uses the retry client to transcribe audio from either a file or your microphone. You can use this application to test the client, or you can use it as a starting point for your own applications.

To run the sample, do the following:

- Copy the retry client to your workspace. See [Streaming Retry Client Code \(p. 27\)](#).
- Copy the retry client interface to your workspace. Implement the interface, or you can use the sample implementation. See [Streaming Retry Client Interface Code \(p. 30\)](#).
- Copy the sample application to your workspace. Build and run the application.

```
package com.amazonaws.wolverine.streaming.retryclient.example;

public class SampleApp {
    private static final String endpoint = "endpoint";
    private static final Region region = region;
    public static void main(String args[]) throws URISyntaxException, ExecutionException,
        InterruptedException, LineUnavailableException, FileNotFoundException {
```

```
/**
 * Create Transcribe streaming retry client using AWS credentials.
 */
TranscribeStreamingRetryClient client = new
TranscribeStreamingRetryClient(getCredentials(), endpoint, region);

StartStreamTranscriptionRequest request =
StartStreamTranscriptionRequest.builder()
    .languageCode(LanguageCode.language.toString())
    .mediaEncoding(encoding)
    .mediaSampleRateHertz(sample rate)
    .build();

/**
 * Start real-time speech recognition. The Transcribe streaming java client uses
the Reactive-streams
 * interface. For reference on Reactive-streams:
 * https://github.com/reactive-streams/reactive-streams-jvm
 */
CompletableFuture<Void> result = client.startStreamTranscription(
    /**
     * Request parameters. Refer to API documentation for details.
     */
    request,
    /**
     * Provide an input audio stream.
     * For input from a microphone, use getStreamFromMic().
     * For input from a file, use getStreamFromFile().
     */
    new AudioStreamPublisher(
        new FileInputStream(new File("FileName"))),
    /**
     * Object that defines the behavior on how to handle the stream
     */
    new StreamTranscriptionBehaviorImpl());

/**
 * Synchronous wait for stream to close, and close client connection
 */
result.get();
client.close();
}

private static class AudioStreamPublisher implements Publisher<AudioStream> {
private final InputStream inputStream;

private AudioStreamPublisher(InputStream inputStream) {
    this.inputStream = inputStream;
}

@Override
public void subscribe(Subscriber<? super AudioStream> s) {
    if (currentSubscription == null) {
        this.currentSubscription = new SubscriptionImpl(s, inputStream);
    } else {
        this.currentSubscription.cancel();
        this.currentSubscription = new SubscriptionImpl(s, inputStream);
    }
    s.onSubscribe(currentSubscription);
}
}
}
```

Monitoring Amazon Transcribe

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Transcribe applications. To monitor Amazon Transcribe, you can use AWS CloudTrail to monitor API calls and the status of your jobs with Amazon CloudWatch Events.

Topics

- [Monitoring Amazon Transcribe API Calls with AWS CloudTrail \(p. 34\)](#)
- [Using Amazon CloudWatch Events with Amazon Transcribe \(p. 36\)](#)

Monitoring Amazon Transcribe API Calls with AWS CloudTrail

Amazon Transcribe is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon Transcribe. CloudTrail captures all API calls for Amazon Transcribe as events, including calls from the Amazon Transcribe console and from code calls to the Amazon Transcribe APIs. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon Transcribe. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Amazon Transcribe, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the [AWS CloudTrail User Guide](#).

Amazon Transcribe Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon Transcribe, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing Events with CloudTrail Event History](#).

For an ongoing record of events in your AWS account, including events for Amazon Transcribe, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all regions. The trail logs events from all regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- [Overview for Creating a Trail](#)
- [CloudTrail Supported Services and Integrations](#)
- [Configuring Amazon SNS Notifications for CloudTrail](#)
- [Receiving CloudTrail Log Files from Multiple Regions](#) and [Receiving CloudTrail Log Files from Multiple Accounts](#)

All Amazon Transcribe actions are logged by CloudTrail and are documented in the [API Reference \(p. 48\)](#). For example, calls to the `CreateVocabulary`, `GetTranscriptionJob` and `StartTranscriptionJob` sections generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the [CloudTrail userIdentity Element](#).

You can also aggregate Amazon Transcribe log files from multiple AWS Regions and multiple AWS accounts into a single S3 bucket. For more information, see [Receiving CloudTrail Log Files from Multiple Regions](#) and [Receiving CloudTrail Log Files from Multiple Accounts](#).

Example: Amazon Transcribe Log File Entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

The following log entry shows the result of calls to the `StartTranscriptionJob` and `GetTranscriptionJob` operations:

```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "AssumedRole | FederatedUser | IAMUser | Root | SAMLUser |
WebIdentityUser",
        "principalId": "principal ID",
        "arn": "ARN",
        "accountId": "account ID",
        "accessKeyId": "access key",
        "userName": "user name"
      },
      "eventTime": "timestamp",
      "eventSource": "transcribe.amazonaws.com",
      "eventName": "StartTranscriptionJob",
      "awsRegion": "region",
      "sourceIPAddress": "source IP address",
      "userAgent": "user agent",
      "requestParameters": {
        "mediaFormat": "flac | mp3 | mp4 | wav",
        "languageCode": "en-US | es-US",
        "transcriptionJobName": "unique job name",
        "media": {
          "mediaFileUri": ""
        }
      },
      "responseElements": {
        "transcriptionJob": {
          "transcriptionJobStatus": "IN_PROGRESS",
          "mediaFormat": "flac | mp3 | mp4 | wav",
          "creationTime": "timestamp",
          "transcriptionJobName": "unique job name",
          "languageCode": "en-US | es-US",
          "media": {
            "mediaFileUri": ""
          }
        }
      },
      "requestID": "request ID",
    }
  ]
}
```

```

        "eventID": "event ID",
        "eventType": "AwsApiCall",
        "recipientAccountId": "account id"
    },
    {
        "eventVersion": "1.05",
        "userIdentity": {
            "type": "AssumedRole | FederatedUser | IAMUser | Root | SAMLUser |
WebIdentityUser",
            "principalId": "principal ID",
            "arn": "ARN",
            "accountId": "account ID",
            "accessKeyId": "access key",
            "userName": "user name"
        },
        "eventTime": "timestamp",
        "eventSource": "transcribe.amazonaws.com",
        "eventName": "GetTranscriptionJob",
        "awsRegion": "region",
        "sourceIPAddress": "source IP address",
        "userAgent": "user agent",
        "requestParameters": {
            "transcriptionJobName": "unique job name"
        },
        "responseElements": {
            "transcriptionJob": {
                "settings": {

                },
                "transcriptionJobStatus": "COMPLETED | FAILED | IN_PROGRESS",
                "mediaFormat": "flac | mp3 | mp4 | wav",
                "creationTime": "timestamp",
                "transcriptionJobName": "unique job name",
                "languageCode": "en-US | es-US",
                "media": {
                    "mediaFileUri": ""
                },
                "transcript": {
                    "transcriptFileUri": ""
                }
            }
        },
        "requestID": "request ID",
        "eventID": "event ID",
        "eventType": "AwsApiCall",
        "recipientAccountId": "account id"
    }
]
}

```

Using Amazon CloudWatch Events with Amazon Transcribe

With Amazon CloudWatch Events, you can respond to state changes in your Amazon Transcribe jobs by triggering events in other AWS services. When a transcription job changes state, CloudWatch Events automatically sends an event to an event stream. You create rules that define the events that you want to monitor in the event stream and the action that CloudWatch Events should take when those events occur, such as routing the event to another service (or target), which can then take an action. For example, you can configure a rule to route an event to a Lambda function when a transcription job is completed successfully.

Before using CloudWatch Events, you should understand the following concepts:

- **Event**—An event indicates a change in the state of one of your transcription jobs. For example, when the `TranscriptionJobStatus` of a job changes from `IN_PROGRESS` to `COMPLETED`.
- **Target**—A target is another AWS service that processes an event, for example, AWS Lambda or Amazon Simple Notification Service. A target receives events in JSON format.
- **Rule**—A rule matches incoming events that you want CloudWatch Events to watch for and routes to a target or targets for processing. A rule can route an event to multiple targets, all of which process the event in parallel. A rule can customize the JSON sent to the target.

For more information about creating and managing CloudWatch Events, see [What is Amazon CloudWatch Events](#) in the *Amazon CloudWatch User Guide*.

Defining CloudWatch Events Rules

Use the [CloudWatch Events console](#) to create CloudWatch Events rules. Define the rule with Amazon Transcribe as the service name. For an example of creating a CloudWatch Events rule, see [Creating a CloudWatch Events Rule That Triggers on an Event](#) in the *Amazon CloudWatch User Guide*.

The following is an example of a CloudWatch Events rule for Amazon Transcribe. This rule is triggered when a transcription job's status changes to `COMPLETED` or `FAILED`.

```
{
  "source": [
    "aws.transcribe"
  ],
  "detail-type": [
    "Transcribe Job State Change"
  ],
  "detail": {
    "TranscriptionJobStatus": [
      "COMPLETED",
      "FAILED"
    ]
  }
}
```

The rule contains the following fields:

- `source`—The source of the event. For Amazon Transcribe, this is always `aws.transcribe`.
- `detail-type`—An identifier for the details of the event. For Amazon Transcribe, this is always `Transcribe Job State Change`.
- `detail`—The new job status of a transcription job. In this example, the rule will raise an event when the job status changes to `COMPLETED` or `FAILED`. For a list of status values, see the `TranscriptionJobStatus` field of the [TranscriptionJob \(p. 83\)](#) data type.

Amazon Transcribe Event

When a job's state changes from `IN_PROGRESS` to either `COMPLETED` or `FAILED`, Amazon Transcribe generates an event. To identify the job that changed state and raised the event in your target, use the event's `TranscriptionJobName` field. An Amazon Transcribe event contains the following information:

```
{
  "version": "0",
  "id": "event ID",
```

```
{
  "detail-type": "Transcribe Job State Change",
  "source": "aws.transcribe",
  "account": "account ID",
  "time": "timestamp",
  "region": "region",
  "resources": [],
  "detail": {
    "TranscriptionJobName": "unique job name",
    "TranscriptionJobStatus": "status"
  }
}
```

The event passed to the target contains the following information:

- **version**—The version of the event data. This value is always 0.
- **id**—A unique identifier generated by CloudWatch Events for the event.
- **detail-type**—An identifier for the details of the event. For Amazon Transcribe, this is always `Transcribe Job State Change`.
- **source**—The source of the event. For Amazon Transcribe this is always `Transcribe Job State Change`.
- **account**—The AWS account ID of the account that generated the API call.
- **timestamp**—The date and time that the API call was made.
- **region**—The AWS Region where the API call was made.
- **resources**—Resources used by the API call. For Amazon Transcribe this field is always empty.
- **detail**—Details about the event. It contains the following fields:
 - **TranscriptionJobName**—The unique job name that you gave the job when it was created.
 - **TranscriptionJobStatus**—The new status of the transcription job. For a list of status values, see the `TranscriptionJobStatus` field of the [TranscriptionJob](#) (p. 83) data type.

Authentication and Access Control for Amazon Transcribe

Access to Amazon Transcribe requires credentials that AWS can use to authenticate your requests. Those credentials must have permissions to access Amazon Transcribe actions. The following sections provide details on how you can use [AWS Identity and Access Management \(IAM\)](#) and Amazon Transcribe to help secure your resources by controlling who can access them.

- [Authentication \(p. 39\)](#)
- [Access Control \(p. 40\)](#)

Authentication

You can access AWS as any of the following types of identities:

- **AWS account root user** – When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the [best practice of using the root user only to create your first IAM user](#). Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.
- **IAM user** – An [IAM user](#) is an identity within your AWS account that has specific custom permissions (for example, permissions to create a custom vocabulary in Amazon Transcribe). You can use an IAM user name and password to sign in to secure AWS webpages like the [AWS Management Console](#), [AWS Discussion Forums](#), or the [AWS Support Center](#).

In addition to a user name and password, you can also generate [access keys](#) for each user. You can use these keys when you access AWS services programmatically, either through [one of the several SDKs](#) or by using the [AWS Command Line Interface \(CLI\)](#). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don't use AWS tools, you must sign the request yourself. Amazon Transcribe supports *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see [Signature Version 4 Signing Process](#) in the *AWS General Reference*.

- **IAM role** – An [IAM role](#) is an IAM identity that you can create in your account that has specific permissions. It is similar to an *IAM user*, but it is not associated with a specific person. An IAM role enables you to obtain temporary access keys that can be used to access AWS services and resources. IAM roles with temporary credentials are useful in the following situations:
- **Federated user access** – Instead of creating an IAM user, you can use existing user identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an [identity](#)

[provider](#). For more information about federated users, see [Federated Users and Roles](#) in the *IAM User Guide*.

- **AWS service access** – You can use an IAM role in your account to grant an AWS service permissions to access your account's resources. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data from that bucket into an Amazon Redshift cluster. For more information, see [Creating a Role to Delegate Permissions to an AWS Service](#) in the *IAM User Guide*.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see [Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances](#) in the *IAM User Guide*.

Access Control

You must have valid credentials to authenticate your requests. The credentials must have permissions to call an Amazon Transcribe action.

The following sections describe how to manage permissions for Amazon Transcribe. We recommend that you read the overview first.

- [Overview of Managing Access Permissions to Amazon Transcribe Resources](#) (p. 40)
- [Using Identity-based Policies \(IAM Policies\) for Amazon Transcribe](#) (p. 42)

Overview of Managing Access Permissions to Amazon Transcribe Resources

Permissions to access an action are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles) to manage access to actions.

Note

An *account administrator* (or administrator user) is a user with administrator privileges. We strongly recommend that you access AWS using IAM not the credentials for your AWS account. To use IAM to access AWS, create an IAM user, add the user to an IAM group with administrative permissions, and then grant administrative permissions to the IAM user. You can then access AWS using a special URL and the IAM user's credentials. For more information, see [IAM Best Practices](#) in the *IAM User Guide*.

When granting permissions, you decide who is getting the permissions and the actions they get permissions for.

Topics

- [Managing Access to Actions](#) (p. 41)
- [Specifying Policy Elements: Resources, Actions, Effects, and Principals](#) (p. 41)
- [Specifying Conditions in a Policy](#) (p. 42)

Managing Access to Actions

A *permissions policy* describes who has access to what. You have several options for creating permissions policies.

Note

This topic discusses using IAM in the context of Amazon Transcribe. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see [What Is IAM?](#) in the *IAM User Guide*. For information about IAM policy syntax and descriptions, see [AWS IAM Policy Reference](#) in the *IAM User Guide*.

Policies attached to an IAM identity are referred to as *identity-based* policies (IAM policies). Policies attached to a resource are referred to as *resource-based* policies. Amazon Transcribe supports only identity-based policies.

Identity-Based Policies (IAM Policies)

You can attach policies to IAM identities. For example, you can do the following:

- **Attach a permissions policy to a user or a group in your account**—To grant a user or a group of users permissions to call an Amazon Transcribe action, you can attach a permissions policy to a user or group that the user belongs to.
- **Attach a permissions policy to a role (grant cross-account permissions)**—To grant cross-account permissions, you can attach an identity-based permissions policy to an IAM role. For example, the administrator in Account A can create a role to grant cross-account permissions to another AWS account (for example, Account B) or an AWS service as follows:
 1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in Account A.
 2. Account A administrator attaches a trust policy to the role identifying Account B as the principal who can assume the role.
 3. Account B administrator can then delegate permissions to assume the role to any users in Account B. Doing this allows users in Account B to create or access resources in Account A. If you want to grant an AWS service permissions to assume the role, the principal in the trust policy can be an AWS service principal.

For more information about using IAM to delegate permissions, see [Access Management](#) in the *IAM User Guide*.

For more information about using identity-based policies with Amazon Transcribe, see [Using Identity-based Policies \(IAM Policies\) for Amazon Transcribe \(p. 42\)](#). For more information about users, groups, roles, and permissions, see [Identities \(Users, Groups, and Roles\)](#) in the *IAM User Guide*.

Resource-Based Policies

Other services, such as AWS Lambda, support resource-based permissions policies. For example, you can attach a policy to an S3 bucket to manage access permissions to that bucket. Amazon Transcribe doesn't support resource-based policies.

Specifying Policy Elements: Resources, Actions, Effects, and Principals

Amazon Transcribe defines a set of API operations (see [Actions \(p. 48\)](#)). To grant permissions for these API operations, Amazon Transcribe defines a set of actions that you can specify in a policy.

The following are the most basic policy elements:

- **Resource**—In a policy, you use an Amazon Resource Name (ARN) to identify the resource to which the policy applies. For Amazon Transcribe, the resource is always `*`.
- **Action**—You use action keywords to identify operations that you want to allow or deny. For example, depending on the specified `Effect`, `transcribe:StartTranscriptionJob` either allows or denies the user permissions to perform the Amazon Transcribe `StartTranscriptionJob` operation.
- **Effect**—You specify the effect of the action that occurs when the user requests the specific action—this can be either allow or deny. If you don't explicitly grant access to (allow) a resource, access is implicitly denied. You can also explicitly deny access to a resource. You might do this to make sure that a user cannot access the resource, even if a different policy grants access.
- **Principal**—In identity-based policies (IAM policies), the user that the policy is attached to is the implicit principal.

To learn more about IAM policy syntax and descriptions, see [AWS IAM Policy Reference](#) in the *IAM User Guide*.

For a table of Amazon Transcribe API actions, see [Amazon Transcribe API Permissions: Actions, Resources, and Conditions Reference](#) (p. 45).

Specifying Conditions in a Policy

When you grant permissions, you use the IAM policy language to specify the conditions under which a policy should take effect. For example, you might want a policy to be applied only after a specific date. For more information about specifying conditions in a policy language, see [Condition](#) in the *IAM User Guide*.

AWS provides a set of predefined condition keys for all AWS services that support IAM for access control. For example, you can use the `aws:userid` condition key to require a specific AWS ID when requesting an action. For more information and a complete list of AWS-wide keys, see [Available Keys for Conditions](#) in the *IAM User Guide*.

Note

Condition keys are case-sensitive.

Amazon Transcribe does not provide additional condition keys.

Using Identity-based Policies (IAM Policies) for Amazon Transcribe

The following identity-based policies demonstrate how an account administrator can attach permissions policies to IAM identities (users, groups, and roles) to grant permissions to perform Amazon Transcribe actions.

Important

Before you proceed, we recommend that you review [Overview of Managing Access Permissions to Amazon Transcribe Resources](#) (p. 40).

The following is the permissions policy required to use the Amazon Transcribe `StartTranscriptionJob` action:

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
{
  "Effect": "Allow",
  "Action": [
    "transcribe:StartTranscriptionJob"
  ],
  "Resource": "*"
}
```

The policy has one statement that grants permissions to use the `StartTranscriptionJob` action.

The policy doesn't specify the `Principal` element because you don't specify the principal who gets the permission in an identity-based policy. When you attach a policy to a user, the user is the implicit principal. When you attach a permissions policy to an IAM role, the principal identified in the role's trust policy gets the permissions.

For a table of Amazon Transcribe API actions and the resources that they apply to, see [Amazon Transcribe API Permissions: Actions, Resources, and Conditions Reference \(p. 45\)](#).

Permissions Required to Use the Amazon Transcribe Console

To use the Amazon Transcribe console, you need to grant permissions for the actions shown in the following policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "transcribe:*"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

The policy enables the user to call all of the Amazon Transcribe operations.

AWS Managed (Predefined) Policies for Amazon Transcribe

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. Managed policies grant necessary permissions for common use cases so you can avoid having to investigate which permissions are needed. For more information, see [AWS Managed Policies](#) in the *IAM User Guide*.

The following AWS managed policies, which you can attach to users in your account, are specific to Amazon Transcribe:

- **ReadOnly** — Grants read-only access to Amazon Transcribe resources so that you can get and list transcription jobs and custom vocabularies.
- **FullAccess** — Grants full access to create, read, update, delete, and run all Amazon Transcribe resources. It also allows access to S3 buckets with "transcribe" in the bucket name.

Note

You can review these permission policies by signing in to the IAM console and searching for specific policies.

You can also create your own custom IAM policies to allow permissions for Amazon Transcribe API actions. You can attach these custom policies to the IAM users or groups that require those permissions.

Permissions Required for IAM User Roles

When you create an IAM user to call Amazon Transcribe, the identity must have permission for the S3 bucket and to the AWS Key Management Service (AWS KMS) key used to encrypt the contents of the bucket, if you provided one.

The user must have the following IAM policy to decrypt permissions on the KMS ARN:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "KMS key ARN",
      "Effect": "Allow"
    }
  ]
}
```

The user's IAM policy must have Amazon S3 permissions to access the S3 bucket where audio files are stored and transcriptions are saved:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
      ],
      "Resource": "S3 bucket location"
    }
  ]
}
```

Permissions Required for Amazon S3 Encryption Keys

If you are using an AWS Key Management Service key to encrypt an Amazon S3 bucket, include the following in the AWS KMS key policy. This allows Amazon Transcribe access to the contents of the bucket. For more information about allowing access to customer master keys, see [Allowing External AWS Accounts to Access a CMK](#) in the *AWS KMS Developer Guide*.

```
{
  "Sid": "Allow-Transcribe",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::account id:root",
  },
  "Action": [
    "kms:Decrypt"
  ]
}
```



```
],  
  "Resource": "KMS key ARN"  
}
```

Amazon Transcribe API Permissions: Actions, Resources, and Conditions Reference

Use the following table as a reference when setting up [Access Control \(p. 40\)](#) and writing a permissions policy that you can attach to an IAM identity (an identity-based policy). The list includes each Amazon Transcribe API operation, the corresponding action for which you can grant permissions to perform the action, and the AWS resource for which you can grant the permissions. You specify the actions in the policy's `Action` field, and you specify the resource value in the policy's `Resource` field.

To express conditions in your Amazon Transcribe policies, you can use AWS-wide condition keys. For a complete list of AWS-wide keys, see [Available Keys](#) in the *IAM User Guide*.

Note

To specify an action, use the `transcribe:` prefix followed by the API operation name, for example, `transcribe:StartTranscriptionJob`.

Guidelines and Limits

Supported Regions

For a list of AWS Regions where Amazon Transcribe is available, see [AWS Regions and Endpoints](#) in the *Amazon Web Services General Reference*.

Throttling

For information about throttling for Amazon Transcribe and to request a limit increase, see [Amazon Transcribe Limits](#) in the *Amazon Web Services General Reference*.

Guidelines

For best results:

- Use a lossless format, such as FLAC or WAV, with PCM 16-bit encoding.
- Use a sample rate of 8000 Hz for low-fidelity audio and 16000 Hz for high-fidelity audio.

Amazon Transcribe may store your content to continuously improve the quality of its analysis models. See the [Amazon Transcribe FAQ](#) to learn more. To request that we delete content that may have been stored by Amazon Transcribe, open a case with AWS Support.

Limits

Amazon Transcribe has the following limitations:

Description	Limit
Maximum audio file length	2 hours
Maximum audio file size	1 GB
Maximum size of a custom vocabulary	50 KB
Maximum length of a custom vocabulary phrase	256 characters

Document History for Amazon Transcribe

- **Latest documentation update:** November 19, 2018

The following table describes important changes in each release of Amazon Transcribe. For notification about updates to this documentation, you can subscribe to an RSS feed.

update-history-change	update-history-description	update-history-date
New feature	Amazon Transcribe adds support for French, Italian, and Brazilian Portuguese.	December 20, 2018
New feature	Amazon Transcribe now supports transcription of audio streams. For more information, see Streaming Transcription .	November 19, 2018
New feature	Amazon Transcribe adds support for Australian English, British English, and Canadian French.	November 15, 2018
Region expansion	Amazon Transcribe is now available in Canada (Central) (ca-central-1) and Asia Pacific (Sydney) (ap-southeast-2).	July 17, 2018
New feature	You can now specify your own location to store the output from a transcription job. For more information, see the TranscriptionJobSummary data type.	July 11, 2018
New feature	Added AWS CloudTrail and Amazon CloudWatch Events integration. For more information, see Monitoring Amazon Transcribe .	June 28, 2018
New feature (p. 47)	Amazon Transcribe adds support for custom vocabularies. For more information, see Create a Custom Vocabulary .	April 4, 2018
New guide (p. 47)	This is the first release of the <i>Amazon Transcribe Developer Guide</i> .	November 29, 2017

API Reference

This section contains the API Reference documentation.

Topics

- [Actions \(p. 48\)](#)
- [Data Types \(p. 77\)](#)

Actions

The following actions are supported by Amazon Transcribe Service:

- [CreateVocabulary \(p. 49\)](#)
- [DeleteTranscriptionJob \(p. 52\)](#)
- [DeleteVocabulary \(p. 54\)](#)
- [GetTranscriptionJob \(p. 56\)](#)
- [GetVocabulary \(p. 58\)](#)
- [ListTranscriptionJobs \(p. 61\)](#)
- [ListVocabularies \(p. 64\)](#)
- [StartTranscriptionJob \(p. 67\)](#)
- [UpdateVocabulary \(p. 71\)](#)

The following actions are supported by Amazon Transcribe Streaming Service:

- [StartStreamTranscription \(p. 74\)](#)

Amazon Transcribe Service

The following actions are supported by Amazon Transcribe Service:

- [CreateVocabulary \(p. 49\)](#)
- [DeleteTranscriptionJob \(p. 52\)](#)
- [DeleteVocabulary \(p. 54\)](#)
- [GetTranscriptionJob \(p. 56\)](#)
- [GetVocabulary \(p. 58\)](#)
- [ListTranscriptionJobs \(p. 61\)](#)
- [ListVocabularies \(p. 64\)](#)
- [StartTranscriptionJob \(p. 67\)](#)
- [UpdateVocabulary \(p. 71\)](#)

CreateVocabulary

Service: Amazon Transcribe Service

Creates a new custom vocabulary that you can use to change the way Amazon Transcribe handles transcription of an audio file.

Request Syntax

```
{  
  "LanguageCode": "string",  
  "Phrases": [ "string" ],  
  "VocabularyName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 98).

The request accepts the following data in JSON format.

LanguageCode (p. 49)

The language code of the vocabulary entries.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT

Required: Yes

Phrases (p. 49)

An array of strings that contains the vocabulary entries.

Type: Array of strings

Length Constraints: Minimum length of 0. Maximum length of 256.

Required: Yes

VocabularyName (p. 49)

The name of the vocabulary. The name must be unique within an AWS account. The name is case-sensitive.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Response Syntax

```
{  
  "FailureReason": "string",  
}
```

```
"LanguageCode": "string",  
"LastModifiedTime": number,  
"VocabularyName": "string",  
"VocabularyState": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

FailureReason (p. 49)

If the `VocabularyState` field is `FAILED`, this field contains information about why the job failed.

Type: String

LanguageCode (p. 49)

The language code of the vocabulary entries.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT

LastModifiedTime (p. 49)

The date and time that the vocabulary was created.

Type: Timestamp

VocabularyName (p. 49)

The name of the vocabulary.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

VocabularyState (p. 49)

The processing state of the vocabulary. When the `VocabularyState` field contains `READY` the vocabulary is ready to be used in a `StartTranscriptionJob` request.

Type: String

Valid Values: `PENDING` | `READY` | `FAILED`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 97\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

ConflictException

When you are using the `StartTranscriptionJob` operation, the `JobName` field is a duplicate of a previously entered job name. Resend your request with a different name.

When you are using the `UpdateVocabulary` operation, there are two jobs running at the same time. Resend the second request later.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

DeleteTranscriptionJob

Service: Amazon Transcribe Service

Deletes a previously submitted transcription job along with any other generated results such as the transcription, models, and so on.

Request Syntax

```
{  
  "TranscriptionJobName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 98\)](#).

The request accepts the following data in JSON format.

TranscriptionJobName (p. 52)

The name of the transcription job to be deleted.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 97\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception Message field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

DeleteVocabulary

Service: Amazon Transcribe Service

Deletes a vocabulary from Amazon Transcribe.

Request Syntax

```
{  
  "VocabularyName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 98\)](#).

The request accepts the following data in JSON format.

VocabularyName (p. 54)

The name of the vocabulary to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 97\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

NotFoundException

We can't find the requested resource. Check the name and try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

GetTranscriptionJob

Service: Amazon Transcribe Service

Returns information about a transcription job. To see the status of the job, check the `TranscriptionJobStatus` field. If the status is `COMPLETED`, the job is finished and you can find the results at the location specified in the `TranscriptionFileUri` field.

Request Syntax

```
{  
  "TranscriptionJobName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 98).

The request accepts the following data in JSON format.

TranscriptionJobName (p. 56)

The name of the job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Response Syntax

```
{  
  "TranscriptionJob": {  
    "CompletionTime": number,  
    "CreationTime": number,  
    "FailureReason": "string",  
    "LanguageCode": "string",  
    "Media": {  
      "MediaFileUri": "string"  
    },  
    "MediaFormat": "string",  
    "MediaSampleRateHertz": number,  
    "Settings": {  
      "ChannelIdentification": boolean,  
      "MaxSpeakerLabels": number,  
      "ShowSpeakerLabels": boolean,  
      "VocabularyName": "string"  
    },  
    "Transcript": {  
      "TranscriptFileUri": "string"  
    },  
    "TranscriptionJobName": "string",  
    "TranscriptionJobStatus": "string"  
  }  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TranscriptionJob (p. 56)

An object that contains the results of the transcription job.

Type: [TranscriptionJob](#) (p. 83) object

Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 97).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

NotFoundException

We can't find the requested resource. Check the name and try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

GetVocabulary

Service: Amazon Transcribe Service

Gets information about a vocabulary.

Request Syntax

```
{  
  "VocabularyName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 98).

The request accepts the following data in JSON format.

VocabularyName (p. 58)

The name of the vocabulary to return information about. The name is case-sensitive.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Response Syntax

```
{  
  "DownloadUri": "string",  
  "FailureReason": "string",  
  "LanguageCode": "string",  
  "LastModifiedTime": number,  
  "VocabularyName": "string",  
  "VocabularyState": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DownloadUri (p. 58)

The S3 location where the vocabulary is stored. Use this URI to get the contents of the vocabulary. The URI is available for a limited time.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2000.

FailureReason (p. 58)

If the VocabularyState field is `FAILED`, this field contains information about why the job failed.

Type: String

LanguageCode (p. 58)

The language code of the vocabulary entries.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT

LastModifiedTime (p. 58)

The date and time that the vocabulary was last modified.

Type: Timestamp

VocabularyName (p. 58)

The name of the vocabulary to return.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

VocabularyState (p. 58)

The processing state of the vocabulary.

Type: String

Valid Values: PENDING | READY | FAILED

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 97\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception Message field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

NotFoundException

We can't find the requested resource. Check the name and try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

ListTranscriptionJobs

Service: Amazon Transcribe Service

Lists transcription jobs with the specified status.

Request Syntax

```
{  
  "JobNameContains": "string",  
  "MaxResults": number,  
  "NextToken": "string",  
  "Status": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 98).

The request accepts the following data in JSON format.

JobNameContains (p. 61)

When specified, the jobs returned in the list are limited to jobs whose name contains the specified string.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

MaxResults (p. 61)

The maximum number of jobs to return in the response. If there are fewer results in the list, this response contains only the actual results.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken (p. 61)

If the result of the previous request to `ListTranscriptionJobs` was truncated, include the `NextToken` to fetch the next set of jobs.

Type: String

Length Constraints: Maximum length of 8192.

Required: No

Status (p. 61)

When specified, returns only transcription jobs with the specified status. Jobs are ordered by creation date, with the newest jobs returned first. If you don't specify a status, Amazon Transcribe returns all transcription jobs ordered by creation date.

Type: String

Valid Values: IN_PROGRESS | FAILED | COMPLETED

Required: No

Response Syntax

```
{
  "NextToken": "string",
  "Status": "string",
  "TranscriptionJobSummaries": [
    {
      "CompletionTime": number,
      "CreationTime": number,
      "FailureReason": "string",
      "LanguageCode": "string",
      "OutputLocationType": "string",
      "TranscriptionJobName": "string",
      "TranscriptionJobStatus": "string"
    }
  ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken (p. 62)

The `ListTranscriptionJobs` operation returns a page of jobs at a time. The maximum size of the page is set by the `MaxResults` parameter. If there are more jobs in the list than the page size, Amazon Transcribe returns the `NextPage` token. Include the token in the next request to the `ListTranscriptionJobs` operation to return in the next page of jobs.

Type: String

Length Constraints: Maximum length of 8192.

Status (p. 62)

The requested status of the jobs returned.

Type: String

Valid Values: IN_PROGRESS | FAILED | COMPLETED

TranscriptionJobSummaries (p. 62)

A list of objects containing summary information for a transcription job.

Type: Array of [TranscriptionJobSummary \(p. 85\)](#) objects

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 97\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

ListVocabularies

Service: Amazon Transcribe Service

Returns a list of vocabularies that match the specified criteria. If no criteria are specified, returns the entire list of vocabularies.

Request Syntax

```
{  
  "MaxResults": number,  
  "NameContains": "string",  
  "NextToken": "string",  
  "StateEquals": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 98).

The request accepts the following data in JSON format.

MaxResults (p. 64)

The maximum number of vocabularies to return in the response. If there are fewer results in the list, this response contains only the actual results.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NameContains (p. 64)

When specified, the vocabularies returned in the list are limited to vocabularies whose name contains the specified string. The search is case-insensitive, `ListVocabularies` will return both "vocabularyname" and "VocabularyName" in the response list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

NextToken (p. 64)

If the result of the previous request to `ListVocabularies` was truncated, include the `NextToken` to fetch the next set of jobs.

Type: String

Length Constraints: Maximum length of 8192.

Required: No

StateEquals (p. 64)

When specified, only returns vocabularies with the `VocabularyState` field equal to the specified state.

Type: String

Valid Values: PENDING | READY | FAILED

Required: No

Response Syntax

```
{
  "NextToken": "string",
  "Status": "string",
  "Vocabularies": [
    {
      "LanguageCode": "string",
      "LastModifiedTime": number,
      "VocabularyName": "string",
      "VocabularyState": "string"
    }
  ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken (p. 65)

The `ListVocabularies` operation returns a page of vocabularies at a time. The maximum size of the page is set by the `MaxResults` parameter. If there are more jobs in the list than the page size, Amazon Transcribe returns the `NextPage` token. Include the token in the next request to the `ListVocabularies` operation to return in the next page of jobs.

Type: String

Length Constraints: Maximum length of 8192.

Status (p. 65)

The requested vocabulary state.

Type: String

Valid Values: IN_PROGRESS | FAILED | COMPLETED

Vocabularies (p. 65)

A list of objects that describe the vocabularies that match the search criteria in the request.

Type: Array of [VocabularyInfo \(p. 87\)](#) objects

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 97\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

StartTranscriptionJob

Service: Amazon Transcribe Service

Starts an asynchronous job to transcribe speech to text.

Request Syntax

```
{
  "LanguageCode": "string",
  "Media": {
    "MediaFileUri": "string"
  },
  "MediaFormat": "string",
  "MediaSampleRateHertz": number,
  "OutputBucketName": "string",
  "Settings": {
    "ChannelIdentification": boolean,
    "MaxSpeakerLabels": number,
    "ShowSpeakerLabels": boolean,
    "VocabularyName": "string"
  },
  "TranscriptionJobName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 98).

The request accepts the following data in JSON format.

LanguageCode (p. 67)

The language code for the language used in the input media file.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT

Required: Yes

Media (p. 67)

An object that describes the input media for a transcription job.

Type: [Media](#) (p. 79) object

Required: Yes

MediaFormat (p. 67)

The format of the input media file.

Type: String

Valid Values: mp3 | mp4 | wav | flac

Required: Yes

MediaSampleRateHertz (p. 67)

The sample rate, in Hertz, of the audio track in the input media file.

Type: Integer

Valid Range: Minimum value of 8000. Maximum value of 48000.

Required: No

OutputBucketName (p. 67)

The location where the transcription is stored.

If you set the `OutputBucketName`, Amazon Transcribe puts the transcription in the specified S3 bucket. When you call the [GetTranscriptionJob \(p. 56\)](#) operation, the operation returns this location in the `TranscriptFileUri` field. The S3 bucket must have permissions that allow Amazon Transcribe to put files in the bucket. For more information, see [Permissions Required for IAM User Roles](#).

Amazon Transcribe uses the default Amazon S3 key for server-side encryption of transcripts that are placed in your S3 bucket. You can't specify your own encryption key.

If you don't set the `OutputBucketName`, Amazon Transcribe generates a pre-signed URL, a shareable URL that provides secure access to your transcription, and returns it in the `TranscriptFileUri` field. Use this URL to download the transcription.

Type: String

Pattern: `[a-z0-9][\.\-a-z0-9]{1,61}[a-z0-9]`

Required: No

Settings (p. 67)

A `Settings` object that provides optional settings for a transcription job.

Type: [Settings \(p. 80\)](#) object

Required: No

TranscriptionJobName (p. 67)

The name of the job. Note that you can't use the strings "." or ".." by themselves as the job name. The name must also be unique within an AWS account.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Response Syntax

```
{
  "TranscriptionJob": {
    "CompletionTime": number,
    "CreationTime": number,
    "FailureReason": "string",
    "LanguageCode": "string",
    "Media": {
      "MediaFileUri": "string"
    }
  },
}
```



```
"MediaFormat": "string",
"MediaSampleRateHertz": number,
"Settings": {
  "ChannelIdentification": boolean,
  "MaxSpeakerLabels": number,
  "ShowSpeakerLabels": boolean,
  "VocabularyName": "string"
},
"Transcript": {
  "TranscriptFileUri": "string"
},
"TranscriptionJobName": "string",
"TranscriptionJobStatus": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TranscriptionJob (p. 68)

An object containing details of the asynchronous transcription job.

Type: [TranscriptionJob \(p. 83\)](#) object

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 97\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception Message field for more information.

HTTP Status Code: 400

ConflictException

When you are using the `StartTranscriptionJob` operation, the `JobName` field is a duplicate of a previously entered job name. Resend your request with a different name.

When you are using the `UpdateVocabulary` operation, there are two jobs running at the same time. Resend the second request later.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

UpdateVocabulary

Service: Amazon Transcribe Service

Updates an existing vocabulary with new values. The `UpdateVocabulary` operation overwrites all of the existing information with the values that you provide in the request.

Request Syntax

```
{
  "LanguageCode": "string",
  "Phrases": [ "string" ],
  "VocabularyName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 98).

The request accepts the following data in JSON format.

LanguageCode (p. 71)

The language code of the vocabulary entries.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT

Required: Yes

Phrases (p. 71)

An array of strings containing the vocabulary entries.

Type: Array of strings

Length Constraints: Minimum length of 0. Maximum length of 256.

Required: Yes

VocabularyName (p. 71)

The name of the vocabulary to update. The name is case-sensitive.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Response Syntax

```
{
  "LanguageCode": "string",
  "LastModifiedTime": number,
  "VocabularyName": "string",
}
```

```
"VocabularyState": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LanguageCode (p. 71)

The language code of the vocabulary entries.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR |
it-IT

LastModifiedTime (p. 71)

The date and time that the vocabulary was updated.

Type: Timestamp

VocabularyName (p. 71)

The name of the vocabulary that was updated.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

VocabularyState (p. 71)

The processing state of the vocabulary. When the `VocabularyState` field contains `READY` the vocabulary is ready to be used in a `StartTranscriptionJob` request.

Type: String

Valid Values: `PENDING` | `READY` | `FAILED`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 97\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

ConflictException

When you are using the `StartTranscriptionJob` operation, the `JobName` field is a duplicate of a previously entered job name. Resend your request with a different name.

When you are using the `UpdateVocabulary` operation, there are two jobs running at the same time. Resend the second request later.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

NotFoundException

We can't find the requested resource. Check the name and try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

Amazon Transcribe Streaming Service

The following actions are supported by Amazon Transcribe Streaming Service:

- [StartStreamTranscription](#) (p. 74)

StartStreamTranscription

Service: Amazon Transcribe Streaming Service

Starts a bidirectional HTTP2 stream where audio is streamed to Amazon Transcribe and the transcription results are streamed to your application.

The following are encoded as HTTP2 headers:

- x-amzn-transcribe-language-code
- x-amzn-transcribe-media-encoding
- x-amzn-transcribe-sample-rate
- x-amzn-transcribe-session-id

Request Syntax

```
POST /stream-transcription HTTP/2
x-amzn-transcribe-language-code: LanguageCode
x-amzn-transcribe-sample-rate: MediaSampleRateHertz
x-amzn-transcribe-media-encoding: MediaEncoding
x-amzn-transcribe-vocabulary-name: VocabularyName
x-amzn-transcribe-session-id: SessionId
Content-type: application/json

{
  "AudioStream": {
    "AudioEvent": {
      "AudioChunk": blob
    }
  }
}
```

URI Request Parameters

The request requires the following URI parameters.

LanguageCode (p. 74)

Indicates the language used in the input audio stream.

Valid Values: en-US

MediaEncoding (p. 74)

The encoding used for the input audio.

Valid Values: pcm

MediaSampleRateHertz (p. 74)

The sample rate, in Hertz, of the input audio. We suggest that you use 8000 Hz for low quality audio and 16000 Hz for high quality audio.

Valid Range: Minimum value of 8000. Maximum value of 48000.

SessionId (p. 74)

A identifier for the transcription session. Use this parameter when you want to retry a session. If you don't provide a session ID, Amazon Transcribe will generate one for you and return it in the response.

Pattern: [a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}

VocabularyName (p. 74)

The name of the vocabulary to use when processing the transcription job.

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Request Body

The request accepts the following data in JSON format.

AudioStream (p. 74)

PCM-encoded stream of audio blobs. The audio stream is encoded as an HTTP2 data frame.

Type: [AudioStream \(p. 91\)](#) object

Required: Yes

Response Syntax

```
HTTP/2 200
x-amzn-request-id: RequestId
x-amzn-transcribe-language-code: LanguageCode
x-amzn-transcribe-sample-rate: MediaSampleRateHertz
x-amzn-transcribe-media-encoding: MediaEncoding
x-amzn-transcribe-vocabulary-name: VocabularyName
x-amzn-transcribe-session-id: SessionId
Content-type: application/json

{
  "TranscriptResultStream": {
    "BadRequestException": {
    },
    "ConflictException": {
    },
    "InternalFailureException": {
    },
    "LimitExceededException": {
    },
    "TranscriptEvent": {
      "Transcript": {
        "Results": [
          {
            "Alternatives": [
              {
                "Items": [
                  {
                    "Content": "string",
                    "EndTime": number,
                    "StartTime": number,
                    "Type": "string"
                  }
                ],
                "Transcript": "string"
              }
            ],
            "EndTime": number,
            "IsPartial": boolean,
            "ResultId": "string",
            "StartTime": number
          }
        ]
      }
    }
  }
}
```

```
}
  }
    }
      ]
        }
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

LanguageCode (p. 75)

The language code for the input audio stream.

Valid Values: `en-US`

MediaEncoding (p. 75)

The encoding used for the input audio stream.

Valid Values: `pcm`

MediaSampleRateHertz (p. 75)

The sample rate for the input audio stream. Use 8000 Hz for low quality audio and 16000 Hz for high quality audio.

Valid Range: Minimum value of 8000. Maximum value of 48000.

RequestId (p. 75)

An identifier for the streaming transcription.

SessionId (p. 75)

An identifier for a specific transcription session.

Pattern: `[a-zA-Z0-9]{8}-[a-zA-Z0-9]{4}-[a-zA-Z0-9]{4}-[a-zA-Z0-9]{4}-[a-zA-Z0-9]{12}`

VocabularyName (p. 75)

The name of the vocabulary used when processing the job.

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

The following data is returned in JSON format by the service.

TranscriptResultStream (p. 75)

Represents the stream of transcription events from Amazon Transcribe to your application.

Type: [TranscriptResultStream \(p. 96\)](#) object

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 97\)](#).

BadRequestException

One or more arguments to the `StartStreamTranscription` operation was invalid. For example, `MediaEncoding` was not set to `pcm` or `LanguageCode` was not set to a valid code. Check the parameters and try your request again.

HTTP Status Code: 400

ConflictException

A new stream started with the same session ID. The current stream has been terminated.

HTTP Status Code: 409

InternalFailureException

A problem occurred while processing the audio. Amazon Transcribe terminated processing. Try your request again.

HTTP Status Code: 500

LimitExceededException

You have exceeded the maximum number of concurrent transcription streams, are starting transcription streams too quickly, or the maximum audio length of 4 hours. Wait until a stream has finished processing, or break your audio stream into smaller chunks and try your request again.

HTTP Status Code: 429

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

Data Types

The following data types are supported by Amazon Transcribe Service:

- [Media](#) (p. 79)
- [Settings](#) (p. 80)
- [Transcript](#) (p. 82)
- [TranscriptionJob](#) (p. 83)
- [TranscriptionJobSummary](#) (p. 85)
- [VocabularyInfo](#) (p. 87)

The following data types are supported by Amazon Transcribe Streaming Service:

- [Alternative](#) (p. 89)
- [AudioEvent](#) (p. 90)
- [AudioStream](#) (p. 91)
- [Item](#) (p. 92)
- [Result](#) (p. 93)
- [Transcript](#) (p. 94)
- [TranscriptEvent](#) (p. 95)
- [TranscriptResultStream](#) (p. 96)

Amazon Transcribe Service

The following data types are supported by Amazon Transcribe Service:

- [Media](#) (p. 79)
- [Settings](#) (p. 80)
- [Transcript](#) (p. 82)
- [TranscriptionJob](#) (p. 83)
- [TranscriptionJobSummary](#) (p. 85)
- [VocabularyInfo](#) (p. 87)

Media

Service: Amazon Transcribe Service

Describes the input media file in a transcription request.

Contents

MediaFileUri

The S3 location of the input media file. The URI must be in the same region as the API endpoint that you are calling. The general form is:

```
https://s3-<aws-region>.amazonaws.com/<bucket-name>/<keyprefix>/<objectkey>
```

For example:

```
https://s3-us-east-1.amazonaws.com/examplebucket/example.mp4
```

```
https://s3-us-east-1.amazonaws.com/examplebucket/mediadocs/example.mp4
```

For more information about S3 object names, see [Object Keys](#) in the *Amazon S3 Developer Guide*.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2000.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

Settings

Service: Amazon Transcribe Service

Provides optional settings for the `StartTranscriptionJob` operation.

Contents

ChannelIdentification

Instructs Amazon Transcribe to process each audio channel separately and then merge the transcription output of each channel into a single transcription.

Amazon Transcribe also produces a transcription of each item detected on an audio channel, including the start time and end time of the item and alternative transcriptions of the item including the confidence that Amazon Transcribe has in the transcription.

You can't set both `ShowSpeakerLabels` and `ChannelIdentification` in the same request. If you set both, your request returns a `BadRequestException`.

Type: Boolean

Required: No

MaxSpeakerLabels

The maximum number of speakers to identify in the input audio. If there are more speakers in the audio than this number, multiple speakers will be identified as a single speaker. If you specify the `MaxSpeakerLabels` field, you must set the `ShowSpeakerLabels` field to true.

Type: Integer

Valid Range: Minimum value of 2. Maximum value of 10.

Required: No

ShowSpeakerLabels

Determines whether the transcription job uses speaker recognition to identify different speakers in the input audio. Speaker recognition labels individual speakers in the audio file. If you set the `ShowSpeakerLabels` field to true, you must also set the maximum number of speaker labels `MaxSpeakerLabels` field.

You can't set both `ShowSpeakerLabels` and `ChannelIdentification` in the same request. If you set both, your request returns a `BadRequestException`.

Type: Boolean

Required: No

VocabularyName

The name of a vocabulary to use when processing the transcription job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

Transcript

Service: Amazon Transcribe Service

Identifies the location of a transcription.

Contents

TranscriptFileUri

The location where the transcription is stored.

Use this URI to access the transcription. If you specified an S3 bucket in the `OutputBucketName` field when you created the job, this is the URI of that bucket. If you chose to store the transcription in Amazon Transcribe, this is a shareable URL that provides secure access to that location.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2000.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

TranscriptionJob

Service: Amazon Transcribe Service

Describes an asynchronous transcription job that was created with the `StartTranscriptionJob` operation.

Contents

CompletionTime

A timestamp that shows when the job was completed.

Type: Timestamp

Required: No

CreationTime

A timestamp that shows when the job was created.

Type: Timestamp

Required: No

FailureReason

If the `TranscriptionJobStatus` field is `FAILED`, this field contains information about why the job failed.

Type: String

Required: No

LanguageCode

The language code for the input speech.

Type: String

Valid Values: `en-US` | `es-US` | `en-AU` | `fr-CA` | `en-GB` | `de-DE` | `pt-BR` | `fr-FR` | `it-IT`

Required: No

Media

An object that describes the input media for the transcription job.

Type: [Media \(p. 79\)](#) object

Required: No

MediaFormat

The format of the input media file.

Type: String

Valid Values: `mp3` | `mp4` | `wav` | `flac`

Required: No

MediaSampleRateHertz

The sample rate, in Hertz, of the audio track in the input media file.

Type: Integer

Valid Range: Minimum value of 8000. Maximum value of 48000.

Required: No

Settings

Optional settings for the transcription job. Use these settings to turn on speaker recognition, to set the maximum number of speakers that should be identified and to specify a custom vocabulary to use when processing the transcription job.

Type: [Settings \(p. 80\)](#) object

Required: No

Transcript

An object that describes the output of the transcription job.

Type: [Transcript \(p. 82\)](#) object

Required: No

TranscriptionJobName

The name of the transcription job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

TranscriptionJobStatus

The status of the transcription job.

Type: String

Valid Values: `IN_PROGRESS` | `FAILED` | `COMPLETED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

TranscriptionJobSummary

Service: Amazon Transcribe Service

Provides a summary of information about a transcription job. .

Contents

CompletionTime

A timestamp that shows when the job was completed.

Type: Timestamp

Required: No

CreationTime

A timestamp that shows when the job was created.

Type: Timestamp

Required: No

FailureReason

If the `TranscriptionJobStatus` field is `FAILED`, a description of the error.

Type: String

Required: No

LanguageCode

The language code for the input speech.

Type: String

Valid Values: `en-US` | `es-US` | `en-AU` | `fr-CA` | `en-GB` | `de-DE` | `pt-BR` | `fr-FR` | `it-IT`

Required: No

OutputLocationType

Indicates the location of the output of the transcription job.

If the value is `CUSTOMER_BUCKET` then the location is the S3 bucket specified in the `outputBucketName` field when the transcription job was started with the `StartTranscriptionJob` operation.

If the value is `SERVICE_BUCKET` then the output is stored by Amazon Transcribe and can be retrieved using the URI in the `GetTranscriptionJob` response's `TranscriptFileUri` field.

Type: String

Valid Values: `CUSTOMER_BUCKET` | `SERVICE_BUCKET`

Required: No

TranscriptionJobName

The name of the transcription job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

TranscriptionJobStatus

The status of the transcription job. When the status is `COMPLETED`, use the `GetTranscriptionJob` operation to get the results of the transcription.

Type: String

Valid Values: `IN_PROGRESS` | `FAILED` | `COMPLETED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

VocabularyInfo

Service: Amazon Transcribe Service

Provides information about a custom vocabulary.

Contents

LanguageCode

The language code of the vocabulary entries.

Type: String

Valid Values: `en-US` | `es-US` | `en-AU` | `fr-CA` | `en-GB` | `de-DE` | `pt-BR` | `fr-FR` | `it-IT`

Required: No

LastModifiedTime

The date and time that the vocabulary was last modified.

Type: Timestamp

Required: No

VocabularyName

The name of the vocabulary.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

VocabularyState

The processing state of the vocabulary. If the state is `READY` you can use the vocabulary in a `StartTranscriptionJob` request.

Type: String

Valid Values: `PENDING` | `READY` | `FAILED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

Amazon Transcribe Streaming Service

The following data types are supported by Amazon Transcribe Streaming Service:

- [Alternative](#) (p. 89)
- [AudioEvent](#) (p. 90)
- [AudioStream](#) (p. 91)
- [Item](#) (p. 92)
- [Result](#) (p. 93)
- [Transcript](#) (p. 94)
- [TranscriptEvent](#) (p. 95)
- [TranscriptResultStream](#) (p. 96)

Alternative

Service: Amazon Transcribe Streaming Service

A list of possible transcriptions for the audio.

Contents

Items

One or more alternative interpretations of the input audio.

Type: Array of [Item \(p. 92\)](#) objects

Required: No

Transcript

The text that was transcribed from the audio.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

AudioEvent

Service: Amazon Transcribe Streaming Service

Provides a wrapper for the audio chunks that you are sending.

Contents

AudioChunk

An audio blob that contains the next part of the audio that you want to transcribe.

Type: Base64-encoded binary data object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

AudioStream

Service: Amazon Transcribe Streaming Service

Represents the audio stream from your application to Amazon Transcribe.

Contents

AudioEvent

A blob of audio from your application. Your audio stream consists of one or more audio events.

Type: [AudioEvent](#) (p. 90) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

Item

Service: Amazon Transcribe Streaming Service

A word or phrase transcribed from the input audio.

Contents

Content

The word or punctuation that was recognized in the input audio.

Type: String

Required: No

EndTime

The offset from the beginning of the audio stream to the end of the audio that resulted in the item.

Type: Double

Required: No

StartTime

The offset from the beginning of the audio stream to the beginning of the audio that resulted in the item.

Type: Double

Required: No

Type

The type of the item. `PRONUNCIATION` indicates that the item is a word that was recognized in the input audio. `PUNCTUATION` indicates that the item was interpreted as a pause in the input audio.

Type: String

Valid Values: `PRONUNCIATION` | `PUNCTUATION`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

Result

Service: Amazon Transcribe Streaming Service

The result of transcribing a portion of the input audio stream.

Contents

Alternatives

A list of possible transcriptions for the audio. Each alternative typically contains one `item` that contains the result of the transcription.

Type: Array of [Alternative \(p. 89\)](#) objects

Required: No

EndTime

The offset in milliseconds from the beginning of the audio stream to the end of the result.

Type: Double

Required: No

IsPartial

`true` to indicate that Amazon Transcribe has additional transcription data to send, `false` to indicate that this is the last transcription result for the audio stream.

Type: Boolean

Required: No

ResultId

A unique identifier for the result.

Type: String

Required: No

StartTime

The offset in milliseconds from the beginning of the audio stream to the beginning of the result.

Type: Double

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

Transcript

Service: Amazon Transcribe Streaming Service

The transcription in a [TranscriptEvent](#) (p. 95).

Contents

Results

[Result](#) (p. 93) objects that contain the results of transcribing a portion of the input audio stream. The array can be empty.

Type: Array of [Result](#) (p. 93) objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

TranscriptEvent

Service: Amazon Transcribe Streaming Service

Represents a set of transcription results from the server to the client. It contains one or more segments of the transcription.

Contents

Transcript

The transcription of the audio stream. The transcription is composed of all of the items in the results list.

Type: [Transcript \(p. 94\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

TranscriptResultStream

Service: Amazon Transcribe Streaming Service

Represents the transcription result stream from Amazon Transcribe to your application.

Contents

BadRequestException

A client error occurred when the stream was created. Check the parameters of the request and try your request again.

Type: Exception
HTTP Status Code: 400

Required: No

ConflictException

A new stream started with the same session ID. The current stream has been terminated.

Type: Exception
HTTP Status Code: 409

Required: No

InternalFailureException

A problem occurred while processing the audio. Amazon Transcribe terminated processing.

Type: Exception
HTTP Status Code: 500

Required: No

LimitExceededException

Your client has exceeded one of the Amazon Transcribe limits, typically the limit on audio length. Break your audio stream into smaller chunks and try your request again.

Type: Exception
HTTP Status Code: 429

Required: No

TranscriptEvent

A portion of the transcription of the audio stream. Events are sent periodically from Amazon Transcribe to your application. The event can be a partial transcription of a section of the audio stream, or it can be the entire transcription of that portion of the audio stream.

Type: [TranscriptEvent](#) (p. 95) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)

- [AWS SDK for Ruby V2](#)

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

InvalidQueryParameter

The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see [Signature Version 4 Signing Process](#) in the *Amazon Web Services General Reference*.

Action

The action to be performed.

Type: string

Required: Yes

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: AWS4-HMAC-SHA256

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: *access_key/YYYYMMDD/region/service/aws4_request*.

For more information, see [Task 2: Create a String to Sign for Signature Version 4](#) in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: 20120325T120000Z.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see [Handling Dates in Signature Version 4](#) in the *Amazon Web Services General Reference*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS Security Token Service, go to [AWS Services That Work with IAM](#) in the *IAM User Guide*.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see [Task 1: Create a Canonical Request For Signature Version 4](#) in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

AWS Glossary

For the latest AWS terminology, see the [AWS Glossary](#) in the *AWS General Reference*.