

- Curl
- Java
- Node
- Python**
- .NET

Overview

- Introduction**
- Endpoint URLs** ▾
- Authentication** ▾
- Error handling**
- Data handling** ▾
- Related information**
- WebSockets** ▾

Methods

- Models** ▾
- Synchronous** ▾
- Asynchronous** ▾
- Custom language models** ▾
- Custom corpora** ▾
- Custom words** ▾

Introduction

Last updated: 2024-06-19

The IBM Watson™ Speech to Text service provides APIs that use IBM's speech-recognition capabilities to produce transcripts of spoken audio. The service can transcribe speech from various languages and audio formats. In addition to basic transcription, the service can produce detailed information about many different aspects of the audio. It returns all JSON response content in the UTF-8 character set.

The service supports three types of models: large speech models that use the locale (ex.: `en-US` , `fr-FR`) as their name, previous-generation models that include the terms `Broadband` and `Narrowband` in their names, and next-generation models that include the terms `Multimedia` and `Telephony` in their names. Broadband and multimedia models have minimum sampling rates of 16 kHz. Narrowband and telephony models have minimum sampling rates of 8 kHz. The large speech models and next-generation models

The code examples on this tab use the client library that is provided for Python.

Installation

```
pip install --upgrade  
"ibm-watson">=7.0.0"
```

GitHub

<https://github.com/watson-developer-cloud/python-sdk>

- Custom grammars ✓
- Custom acoustic models ✓
- Custom audio resources ✓
- User data ✓

offer high throughput and greater transcription accuracy.

Deprecated: Effective **31 July 2023**, all previous-generation models will be removed from the service and the documentation. Most previous-generation models were deprecated on 15 March 2022. You must migrate to the equivalent large speech model or next-generation model by 31 July 2023. For more information, see [Migrating to large speech models](#)

For speech recognition, the service supports synchronous and asynchronous HTTP Representational State Transfer (REST) interfaces. It also supports a WebSocket interface that provides a full-duplex, low-latency communication channel: Clients send requests and audio to the service and receive results over a single connection asynchronously.

The service also offers two customization interfaces. Use language model customization to expand the vocabulary of a base model with domain-specific terminology. Use acoustic model customization to adapt a base model for

the acoustic characteristics of your audio. For language model customization, the service also supports grammars. A grammar is a formal language specification that lets you restrict the phrases that the service can recognize.

Language model customization is available for most large speech models, previous- and next-generation models. Acoustic model customization is available for all previous-generation models.

Important: This documentation describes Python SDK major version 5. For more information about how to update your code from the previous version, see the [migration guide](#) .

Endpoint URLs

Identify the base URL for your service instance.

IBM Cloud URLs

The following URLs represent the base URLs for Speech to Text. When you call the API, use the URL that corresponds to the location of your service instance.

- Dallas: <https://api.us-south.speech-to-text.watson.cloud.ibm.com>
- Washington, DC: <https://api.us-east.speech-to-text.watson.cloud.ibm.com>
- Frankfurt: <https://api.eu-de.speech-to-text.watson.cloud.ibm.com>
- Sydney: <https://api.au-syd.speech-to-text.watson.cloud.ibm.com>
- Tokyo: <https://api.jp-tok.speech-to-text.watson.cloud.ibm.com>
- London: <https://api.eu-gb.speech-to-text.watson.cloud.ibm.com>
- Seoul: <https://api.kr-seo.speech-to-text.watson.cloud.ibm.com>

Set the correct service URL by calling the `set_service_url()` method of the service instance.

Default URL

<https://api.us-south.speech-to-text.watson.cloud.ibm.com>

Example for the Washington, DC location

```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('https://api.us-
east.speech-to-
text.watson.cloud.ibm.com'
)
```

Cloud Pak for Data URLs

For services installed on Cloud Pak for Data, the base URLs come from both the cluster and service instance.

You can find the base URL from the Cloud Pak for Data web client in the details page about the instance. Click the name of the service in your list of instances to see the URL.

Set the correct service URL by specifying the `url` parameter when you create the service instance or by calling the `set_url()` method of the service instance. For Cloud Pak for Data System, use a hostname that resolves to an IP address in the cluster.

Endpoint example for Cloud Pak for Data

Disabling SSL verification

```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
CloudPakForDataAuthenticat
or

authenticator =
CloudPakForDataAuthenticat
or(
    '{username}',
    '{password}',

    'https://{cpd_cluster_host}
    {port}'
)

speech_to_text =
SpeechToTextV1(

    authenticator=authenticato
    r
)

speech_to_text.set_service
_url('https://{cpd_cluster
_host}{port}/speech-to-
text/{deployment_id}/insta
nces/{instance_id}/api')
```

All Watson services use Secure Sockets Layer (SSL) (or Transport Layer Security (TLS)) for secure connections between the client and server. The connection is verified against the local certificate store to ensure authentication, integrity, and confidentiality.

If you use a self-signed certificate, you need to disable SSL verification to make a successful connection.

Important: Enabling SSL verification is highly recommended. Disabling SSL jeopardizes the security of the connection and data. Disable SSL only if necessary, and take steps to enable SSL as soon as possible.

To disable SSL verification, specify `True` on the `set_disable_ssl_verification` method for the service instance.

Example to disable SSL verification with a service managed on IBM Cloud

```
from ibm_watson import  
SpeechToTextV1  
from  
ibm_cloud_sdk_core.authent  
icators import  
IAMAuthenticator
```

```
authenticator =  
IAMAuthenticator('{apikey}  
' )  
speech_to_text =  
SpeechToTextV1(
```

```
authenticator=authenticato  
r  
)
```

```
speech_to_text.set_service  
_url('{url}')
```

```
speech_to_text.set_disable  
_ssl_verification(True)
```

Example to disable SSL verification with an installed service

Authentication

```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
CloudPakForDataAuthenticat
or

authenticator =
CloudPakForDataAuthenticat
or(
    '{username}',
    '{password}'
)

speech_to_text =
SpeechToTextV1(

    authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

speech_to_text.set_disable
_ssl_verification(True)
```


IBM Cloud services use IBM Cloud Identity and Access Management (IAM) to authenticate. With IBM Cloud Pak for Data, you pass a bearer token.

For IBM Cloud instances, you authenticate to the API by using IBM Cloud Identity and Access Management (IAM).

You can pass either a bearer token in an authorization header or an API key. Tokens support authenticated requests without embedding service credentials in every call. API keys use basic authentication. For more information, see [Authenticating to Watson services](#).

For IBM Cloud instances, the SDK provides initialization methods for each form of authentication.

- Use the API key to have the SDK manage the lifecycle of the access token. The SDK requests an access token, ensures that the access token is valid, and refreshes it if necessary.
- Use the access token to manage the lifecycle yourself. You must periodically refresh the token.

For more information, see [IAM authentication](#) with the SDK.

IBM Cloud. SDK managing the IAM token. Replace `{apikey}` and `{url}`.

```
from ibm_watson import 
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')
```

Cloud Pak for Data

For Cloud Pak for Data instances, pass either username and password credentials or a bearer token that you generate to authenticate to the API. Username and password credentials use basic authentication. However, the SDK manages the lifecycle of the token. Tokens are temporary security credentials. If you pass a token, you maintain the token lifecycle.

For production use, create a user in the Cloud Pak for Data web client to use for authentication, and decide which authentication mechanism to use.

- To have the SDK manage the lifecycle of the token, use the username and password for that new user in your calls.
- To manage the lifecycle of the token yourself, generate a token from that user's credentials. Call the `POST /v1/authorize` method to generate the token, and then pass the token in an `Authorization` header in your calls. You can see an example of the method on the [Curl](#) tab.

For more information, see the **Get authorization token** method of the [Cloud Pak for Data API reference](#) .

Don't use the bearer token that's displayed in

Cloud Pak for Data. SDK managing the token.


Replace `{username}` and `{password}` with your Cloud Pak for Data credentials. For `{url}`, see [Endpoint URLs](#).

the web client for the instance except during testing and development because that token does not expire.

Tip: To find your value for `{url}`, view the details for the service instance by clicking the name of the service in your list of instances in the Cloud Pak for Data web client.

Access between services

Your application might use more than one Watson service. You can grant access between



```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
CloudPakForDataAuthenticat
or

authenticator =
CloudPakForDataAuthenticat
or(
    '{username}',
    '{password}',

    'https://{cpd_cluster_host}
:{port}/icp4d-
api/v1/authorize'
)

speech_to_text =
SpeechToTextV1(

    authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')
```

services and you can grant access to more than one service for your applications.

For IBM Cloud services, the method to grant access between Watson services varies depending on the type of API key. For more information, see [IAM access](#) .

- To grant access between IBM Cloud services, create an authorization between the services. For more information, see [Granting access between services](#) .
- To grant access to your services by applications without using user credentials, create a service ID, add an API key, and assign access policies. For more information, see [Creating and working with service IDs](#) .

When you give a user ID access to multiple services, use an endpoint URL that includes the service instance ID (for example, <https://api.us-south.speech-to-text.watson.cloud.ibm.com/instances/6bbda3b3-d572-45e1-8c54-22d6ed9e52c2>). You can find the instance ID in two places:

- By clicking the service instance row in the [Resource list](#) . The instance ID is the **GUID** in the details pane.
- By clicking the name of the service instance in the list and looking at the credentials URL.

Tip: If you don't see the instance ID in the URL, the credentials predate service IDs. Add new credentials from the **Service credentials** page and use those credentials.

Because the Cloud Pak for Data bearer token is associated with a username, you can use the token for all CPD Watson services that are associated with the username.

Error handling

Speech to Text uses standard HTTP response codes to indicate whether a method completed successfully. HTTP response codes in the 2xx range indicate success. A response in the 4xx range is some sort of failure, and a response in the 5xx range usually indicates an internal system error that cannot be resolved by the user. Response codes are listed with the method.

The Python SDK generates an exception for any unsuccessful method invocation. When the Python SDK receives an error response from the Speech to Text service, it generates an `ApiException` with the following fields.

Field	Description
code	The HTTP response code that is returned.
message	A message that describes the error.
info	A dictionary of additional information about the error.

Data handling

Example error handling

```
from ibm_watson import  
ApiException  
try:  
    # Invoke a method  
except ApiException as ex:  
    print "Method failed  
with status code " +  
str(ex.code) + ": " +  
ex.message
```

Additional headers

Some Watson services accept special parameters in headers that are passed with the request.

You can pass request header parameters in all requests or in a single request to the service.

To pass header parameters with every request, specify the `set_default_headers` method of the service object. See [Data collection](#) for an example use of this method.

To pass header parameters in a single request, include `headers` as a `dict` in the request.

Response details

The Speech to Text service might return information to the application in response headers.

Example header parameter in a request

```
response =  
speech_to_text.methodName(  
    parameters,  
    headers = {  
        'Custom-Header':  
'{header_value}'  
    })
```

The return value from all service methods is a `DetailedResponse` object. To access information in the result object or response headers, use the following methods.

DetailedResponse

Method	Description
<code>get_result()</code>	Returns the response for the service-specific method.
<code>get_headers()</code>	Returns the response header information.
<code>get_status_code()</code>	Returns the HTTP status code.

Data labels (IBM Cloud)

Example request to access response headers

```
speech_to_text.set_detail_response(True)
response = speech_to_text.methodName(parameters)
# Access response from methodName
print(json.dumps(response.get_result(), indent=2))
# Access information in response headers
print(response.get_headers())
# Access HTTP response status
print(response.get_status_code())
```


You can remove data associated with a specific customer if you label the data with a customer ID when you send a request to the service.

- Use the `X-Watson-Metadata` header to associate a customer ID with the data. By adding a customer ID to a request, you indicate that it contains data that belongs to that customer.

Specify a random or generic string for the customer ID. Do not include personal data, such as an email address. Pass the string `customer_id={id}` as the argument of the header.

Labeling data is used only by methods that accept customer data.

- Use the **Delete labeled data** method to remove data that is associated with a customer ID.

Use this process of labeling and deleting data only when you want to remove the data that is associated with a single customer, not when you want to remove data for multiple customers. For more information about Speech to Text and labeling data, see [Information security](#).

For more information about how to pass headers, see [Additional headers](#).

Data collection (IBM Cloud)

By default, Speech to Text service instances managed on IBM Cloud that are not part of Premium plans collect data about API requests and their results. This data is collected only to improve the services for future users. The collected data is not shared or made public. Data is not collected for services that are part of Premium plans.

To prevent IBM usage of your data for an API request, set the **X-Watson-Learning-Opt-Out** header parameter to `true`. You can also disable request logging at the account level. For more information, see [Controlling request logging for Watson services](#).

You can set the header by using the `set_default_headers` method of the service object.

Related information

- Speech to Text [docs](#)
- [Release notes for IBM Cloud](#)
- [Release notes for IBM Cloud Pak for Data](#)

WebSockets

Example request with a service managed on IBM Cloud

```
speech_to_text.set_default_headers({'x-watson-learning-opt-out':  
"true"})
```



Sends audio and returns transcription results for recognition requests over a WebSocket connection. Requests and responses are enabled over a single TCP connection that abstracts much of the complexity of the request to offer efficient implementation, low latency, high throughput, and an asynchronous response.

You can pass a maximum of 100 MB and a minimum of 100 bytes of audio per recognition request. You can send multiple requests over a single WebSocket connection. The service automatically detects the endianness of the incoming audio and, for audio that includes multiple channels, downmixes the audio to one-channel mono during transcoding.

By default, the service returns only final results for any request. You can request interim results to see intermediate hypotheses as the transcription progress.

See also:

- [The WebSocket interface](#)
- [How the service sends recognition results](#)
- [Interim results](#)
- [Requesting interim results and low latency](#)

Deprecated: The

`recognize_with_websocket`

method is deprecated. Use the equivalent

`recognize_using_websocket`

method instead.

Audio formats (content types)

The service accepts audio in the following formats (MIME types).

- For formats that are labeled **Required**, you must use the `content_type` parameter with the request to specify the format of the audio.
- For all other formats, you can omit the `content_type` parameter or specify `application/octet-stream` with the parameter to have the service automatically detect the format of the audio.

Where indicated, the format that you specify must include the sampling rate and can optionally include the number of channels and the endianness of the audio.

- `application/octet-stream`
- `audio/alaw` (**Required**. Specify the sampling rate (`rate`) of the audio.)
- `audio/basic` (**Required**. Use only with

narrowband models.)

- `audio/flac`
- `audio/g729` (Use only with narrowband models.)
- `audio/l16` (**Required.** Specify the sampling rate (`rate`) and optionally the number of channels (`channels`) and endianness (`endianness`) of the audio.)
- `audio/mp3`
- `audio/mpeg`
- `audio/mulaw` (**Required.** Specify the sampling rate (`rate`) of the audio.)
- `audio/ogg` (The service automatically detects the codec of the input audio.)
- `audio/ogg;codecs=opus`
- `audio/ogg;codecs=vorbis`
- `audio/wav` (Provide audio with a maximum of nine channels.)
- `audio/webm` (The service automatically detects the codec of the input audio.)
- `audio/webm;codecs=opus`
- `audio/webm;codecs=vorbis`

See also:

- [Supported audio formats](#)

Important: The Python


`recognize_using_websocket`
method requires the
`content_type` parameter.

Large speech models and Next-generation models

The service supports large speech models and next-generation `Multimedia` (16 kHz) and `Telephony` (8 kHz) models for many languages. Large speech models and next-generation models have higher throughput than the service's previous generation of `Broadband` and `Narrowband` models. When you use large speech models and next-generation models, the service can return transcriptions more quickly and also provide noticeably better transcription accuracy.

You specify a large speech model or next-generation model by using the `model` parameter, as you do a previous-generation model. Only the next-generation models support the `low_latency` parameter, and all large speech models and next-generation models support the `character_insertion_bias` parameter. These parameters are not available with previous-generation models.

Large speech models and next-generation



```
dict
recognize_using_websocket(
    audio, content_type,
    recognize_callback,
    model=None,

    language_customization_id=
    None,
    acoustic_customization_id=
    None,

    customization_weight=None,
    base_model_version=None,
    inactivity_timeout=None,
    interim_results=None,
    keywords=None,
    keywords_threshold=None,
    max_alternatives=None,
    word_alternatives_threshol
    d=None,
    word_confidence=None,
    timestamps=None,
    profanity_filter=None,
    smart_formatting=None,
    speaker_labels=None,
```

models do not support all of the speech recognition parameters that are available for use with previous-generation models. Next-generation models do not support the following parameters:

- `acoustic_customization_id`
- `keywords` and `keywords_threshold`
- `processing_metrics` and `processing_metrics_interval`
- `word_alternatives_threshold`

Deprecated: Effective **31 July 2023**, all previous-generation models will be removed from the service and the documentation. Most previous-generation models were deprecated on 15 March 2022. You must migrate to the equivalent large speech model or next-generation model by 31 July 2023. For more information, see [Migrating to large speech models](#).

See also:

- [Large speech languages and models](#)
- [Supported features for large speech models](#)

```
http_proxy_host=None,  
    http_proxy_port=None,  
    customization_id=None,  
    grammar_name=None,  
    redaction=None,  
    processing_metrics=None,  
    processing_metrics_interval=None,  
    audio_metrics=None,  
    end_of_phrase_silence_time=None,  
  
    split_transcript_at_phrase_end=None,  
    speech_detector_sensitivity=None,  
  
    background_audio_suppression=None, **kwargs)
```

- [Next-generation languages and models](#)
- [Supported features for next-generation models](#)

Request

Call the `recognize_using_websocket` method to initiate a recognition request. Pass the audio and all parameters of the request, including the `RecognizeCallback` and `AudioSource` objects, as arguments of the method.

Parameters	
audio Required [*] AudioSource	An <code>AudioSource</code> object that provides the audio that is to be transcribed.

Example request

```
import json
from os.path import join,
dirname
from ibm_watson import
SpeechToTextV1
from ibm_watson.websocket
import RecognizeCallback,
AudioSource

#####

# IBM CLOUD: Use the
following code only to
# authenticate to IBM
Cloud.
#####

from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator
authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(
```


content_type The format (MIME type) of the audio. For more information about specifying an audio format, see **Audio formats (content types)** in the method description.

Required *
str

Allowable values:

[[application/octet-stream](#), [audio/alaw](#), [audio/basic](#), [audio/flac](#), [audio/g729](#), [audio/l16](#), [audio/mp3](#), [audio/mpeg](#), [audio/mulaw](#), [audio/ogg](#), [audio/ogg;codecs=opus](#), [audio/ogg;codecs=vorbis](#), [audio/wav](#), [audio/webm](#), [audio/webm;codecs=opus](#), [audio/webm;codecs=vorbis](#)]

```
authenticator=authenticator
)
speech_to_text.set_service_url('{url}')
```

```
#####
```

```
# IBM CLOUD PAK FOR DATA:
Use the following code
# only to authenticate to
IBM Cloud Pak for Data.
```

```
#####
```

```
# from
ibm_cloud_sdk_core.authenticators import
CloudPakForDataAuthenticator
or
# authenticator =
CloudPakForDataAuthenticator(
#     '{username}',
#     '{password}',
#     'https://{cpd_cluster_host}:{port}'
# )
# speech_to_text =
SpeechToTextV1(
#
authenticator=authenticator
# )
```

recognize_callback

Required *

object

A `RecognizeCallback`

object that defines methods to handle events from the WebSocket connection. Override the definitions of the object's default methods to respond to events as needed by your application.

model

str

The model to use for all speech recognition requests that are sent over the connection. See [Using a model for speech recognition](#) .

The default model is `en-US_BroadbandModel` .

For Speech to Text for IBM Cloud Pak for Data, if you do not install the `en-US_BroadbandModel` , you must either specify a model with the request or specify a new default model for your installation

```
#  
speech_to_text.set_service  
_url('{url}')
```

```
class  
MyRecognizeCallback(Recogn  
izeCallback):  
    def __init__(self):
```

```
RecognizeCallback.__init__  
(self)
```

```
    def on_data(self,  
data):
```

```
print(json.dumps(data,  
indent=2))
```

```
    def on_error(self,  
error):  
        print('Error  
received:  
{}`.format(error))
```

```
    def  
on_inactivity_timeout(self  
, error):  
        print('Inactivity  
timeout:  
{}`.format(error))
```

```
myRecognizeCallback =  
MyRecognizeCallback()
```

of the service. For more information, see [Using the default model](#).

Allowable values: [ar-MS_BroadbandModel , ar-MS_Telephony , cs-CZ_Telephony , de-DE , de-DE_BroadbandModel , de-DE_Multimedia , de-DE_NarrowbandModel , de-DE_Telephony , en-AU , en-AU_BroadbandModel , en-AU_Multimedia , en-AU_NarrowbandModel , en-AU_Telephony , en-GB , en-GB_BroadbandModel , en-GB_Multimedia , en-GB_NarrowbandModel , en-GB_Telephony , en-IN , en-IN_Telephony , en-US , en-US_BroadbandModel , en-US_Multimedia , en-US_NarrowbandModel ,

```
with
open(join(dirname(__file__
), './.', 'audio-
file.flac'),
      'rb') as
audio_file:
    audio_source =
    AudioSource(audio_file)

speech_to_text.recognize_u
sing_websocket(

audio=audio_source,

content_type='audio/flac',

recognize_callback=myRecog
nizeCallback,
    model='en-
US_BroadbandModel',
    keywords=
['colorado', 'tornado',
'tornadoes'],

keywords_threshold=0.5,

max_alternatives=3)
```

en-
US_ShortForm_NarrowbandModel , en-
US_Telephony , en-
WW_Medical_Telephony
, es-AR , es-
AR_BroadbandModel ,
es-
AR_NarrowbandModel ,
es-CL , es-
CL_BroadbandModel ,
es-
CL_NarrowbandModel ,
es-CO , es-
CO_BroadbandModel ,
es-
CO_NarrowbandModel ,
es-ES , es-
ES_BroadbandModel ,
es-
ES_NarrowbandModel ,
es-ES_Multimedia ,
es-ES_Telephony , es-
LA_Telephony , es-MX ,
es-
MX_BroadbandModel ,
es-
MX_NarrowbandModel ,
es-PE , es-
PE_BroadbandModel ,
es-

PE_NarrowbandModel ,
fr-CA , fr-
CA_BroadbandModel ,
fr-CA_Multimedia ,
fr-
CA_NarrowbandModel ,
fr-CA_Telephony , fr-
FR , fr-
FR_BroadbandModel ,
fr-FR_Multimedia ,
fr-
FR_NarrowbandModel ,
fr-FR_Telephony , hi-
IN_Telephony , it-
IT_BroadbandModel ,
it-
IT_NarrowbandModel ,
it-IT_Multimedia ,
it-IT_Telephony , ja-
JP , ja-
JP_BroadbandModel ,
ja-JP_Multimedia ,
ja-
JP_NarrowbandModel ,
ja-JP_Telephony , ko-
KR_BroadbandModel ,
ko-KR_Multimedia ,
ko-
KR_NarrowbandModel ,
ko-KR_Telephony , nl-
BE_Telephony , nl-

```
NL_BroadbandModel ,  
  nl-NL_Multimedia ,  
  nl-  
NL_NarrowbandModel ,  
  nl-NL_Telephony , pt-  
BR , pt-  
BR_BroadbandModel ,  
  pt-BR_Multimedia ,  
  pt-  
BR_NarrowbandModel ,  
  pt-BR_Telephony , sv-  
SE_Telephony , zh-  
CN_BroadbandModel ,  
  zh-  
CN_NarrowbandModel ,  
  zh-CN_Telephony ]  
  
Default: en-  
US_BroadbandModel
```

language_customization_id

str

The customization ID (GUID) of a custom language model that is to be used for the request.

The base model of the specified custom language model must match the model that is specified with the `model` parameter. You must make the request with credentials for the instance of the service that owns the custom model.

Omit the parameter to use the specified model with no custom language model.

See [Using a custom language model for speech recognition](#) .

acoustic_customization_id

str

The customization ID (GUID) of a custom acoustic model that is to be used for the request. The base model of the specified custom acoustic model must match the model that is specified with the `model` parameter. You must make the request with credentials for the instance of the service that owns the custom model. Omit the parameter to use the specified model with no custom acoustic model. See [Using a custom acoustic model for speech recognition](#) .

customization_weight

float

If you specify a customization ID, you can use the customization weight to tell the service how much weight to give to words from the custom

language model compared to those from the base model for the current request.

Specify a value between 0.0 and 1.0. Unless a different customization weight was specified for the custom model when the model was trained, the default value is:

- 0.5 for large speech models
- 0.3 for previous-generation models
- 0.2 for most next-generation models
- 0.1 for next-generation English and Japanese models

A customization weight that you specify overrides a weight that was specified when the custom model was trained. The default value yields the best performance in general. Assign a higher value if your audio makes frequent

use of OOV words from the custom model. Use caution when you set the weight: a higher value can improve the accuracy of phrases from the custom model's domain, but it can negatively affect performance on non-domain phrases.

See [Using customization weight](#) .

base_model_version

str

The version of the specified base model that is to be used for the request.

Multiple versions of a base model can exist when a model is updated for internal improvements.

The parameter is intended primarily for use with custom models that have been upgraded for a new base model. The default value depends on whether the parameter is used with or without a custom model.

See [Making speech recognition requests with upgraded custom models](#)

inactivity_timeout

int

The time in seconds after which, if only silence (no speech) is detected in the audio, the connection is closed. The default is 30 seconds. The parameter is useful for stopping audio submission from a live microphone when a user simply walks away. Use `-1` for infinity. See

[Inactivity timeout](#) .

Default: `30`

interim_results If `true` , the service

boolean bool

returns intermediate

hypotheses as a stream of JSON

`SpeechRecognitionResults` objects before

returning final results for an utterance. If `false` , the service returns only a single

`SpeechRecognitionResults` object with final results for any utterance.

- *For previous-generation*

models, interim results are available for all models. To receive interim results, set the `interim_results` parameter to `true` .

- *For next-generation models*, interim results are available only for those models that support low latency. To receive interim results, see both the `interim_results` and `low_latency` parameters to `true` .

For for information, see:

- [Interim results](#)
- [Requesting interim results and low latency](#)
- [How the service sends recognition results](#)

Default: `false`

keywords`list[str]`

An array of keyword strings to spot in the audio. Each keyword string can include one or more string tokens. Keywords are spotted only in the final results, not in interim hypotheses. If you specify any keywords, you must also specify a keywords threshold. Omit the parameter or specify an empty array if you do not need to spot keywords.

You can spot a maximum of 1000 keywords with a single request. A single keyword can have a maximum length of 1024 characters, though the maximum effective length for double-byte languages might be shorter. Keywords are case-insensitive.

See [Keyword spotting](#) .

keywords_threshold

float

A confidence value that is the lower bound for spotting a keyword. A word is considered to match a keyword if its confidence is greater than or equal to the threshold. Specify a probability between 0.0 and 1.0. No keyword spotting is performed if you omit the parameter. If you specify a threshold, you must also specify one or more keywords. See [Keyword spotting](#) .

max_alternatives

int

The maximum number of alternative transcripts that the service is to return. By default, the service returns a single transcript. If you specify a value of `0`, the service uses the default value, `1`. See [Maximum alternatives](#).

Default: `1`

word_alternatives_threshold

float

A confidence value that is the lower bound for identifying a hypothesis as a possible word alternative (also known as "Confusion Networks"). An alternative word is considered if its confidence is greater than or equal to the threshold. Specify a probability between 0.0 and 1.0. By default, the service computes no alternative words. See [Word alternatives](#) .

word_confidence

bool

If **true** , the service returns a confidence measure in the range of 0.0 to 1.0 for each word. By default, no word confidence measures are returned. See [Word confidence](#) .

Default: **false**

timestamps

bool

If `true`, the service returns time alignment for each word. By default, no timestamps are returned. See [Word timestamps](#).

Default: `false`

profanity_filter

bool

If `true`, the service filters profanity from all output except for keyword results by replacing inappropriate words with a series of asterisks. Set the parameter to `false` to return results with no censoring.

Note: The parameter can be used with US English and Japanese transcription only. See [Profanity filtering](#).

Default: `true`

smart_formatting

bool

If `true`, the service converts dates, times, series of digits and numbers, phone numbers, currency values, and internet addresses into more readable, conventional representations in the final transcript of a recognition request. For US English, the service also converts certain keyword strings to punctuation symbols. By default, no smart formatting is performed.

Beta: The parameter is beta functionality. It can be used with US English, Japanese, and Spanish (all dialects) transcription only. See [Smart formatting](#).

Default: `false`

smart_formatting_version

int

Smart formatting version

for large speech models

and next-generation

models is supported in US

English, Brazilian

Portuguese, French,

German, Spanish and

French Canadian

languages.

See [Smart formatting](#)

[Version](#) .

Default: 0

speaker_labels If `true` , the response
bool includes labels that
identify which words were
spoken by which
participants in a multi-
person exchange. By
default, the service returns
no speaker labels. Setting
`speaker_labels` to
`true` forces the
`timestamps` parameter
to be `true` , regardless of
whether you specify
`false` for the parameter.

- *For previous-generation*

models, can be used with Australian English, US English, German, Japanese, Korean, and Spanish (both broadband and narrowband models) and UK English (narrowband model) transcription only.

- *For large speech models and next-generation models*, can be used with all available languages.

See [Speaker labels](#) .

Default: `false`

http_proxy_host

str

If you are passing requests through a proxy, specify the hostname of the proxy server. Use the

[http_proxy_port](#)

parameter to specify the port number at which the proxy listens. Omit both parameters if you are not using a proxy.

Default: [None](#)

http_proxy_port

str

If you are passing requests through a proxy, specify the port number at which the proxy service listens.

Use the

[http_proxy_host](#)

parameter to specify the hostname of the proxy.

Omit both parameters if you are not using a proxy.

Default: [None](#)

grammar_name str	The name of a grammar that is to be used with the recognition request. If you specify a grammar, you must also use the <code>language_customization_id</code> parameter to specify the name of the custom language model for which the grammar is defined. The service recognizes only strings that are recognized by the specified grammar; it does not recognize other custom words from the model's words resource.
----------------------------	--

redaction bool	If <code>true</code> , the service redacts, or masks, numeric data from final transcripts. The feature redacts any number that has three or more consecutive digits by replacing each digit with an <code>X</code> character. It is intended to redact sensitive numeric data, such as credit card numbers. By default, the service performs no redaction.
--------------------------	--

When you enable redaction, the service automatically enables smart formatting, regardless of whether you explicitly disable that feature. To ensure maximum security, the service also disables keyword spotting (ignores the `keywords` and `keywords_threshold` parameters) and returns only a single final transcript (forces the `max_alternatives` parameter to be `1`).

Beta: The parameter is beta functionality. It can be used with US English, Japanese, and Korean transcription only. See [Numeric redaction](#) .

Default: `false`

processing_metrics

bool

If `true` , requests processing metrics about the service's transcription of the input audio. The service returns processing metrics at the interval that is specified by the `processing_metrics_interval` parameter. It also returns processing metrics for transcription events, for example, for final and interim results. By default, the service returns no processing metrics. See [Processing metrics](#) .

Default: `false`

processing_metrics_interval

float

Specifies the interval in seconds at which the service is to return processing metrics. The parameter is ignored unless the `processing_metrics` parameter is set to `true` .

The parameter accepts a minimum value of 0.1 seconds. The level of precision is not restricted, so you can specify values such as 0.25 and 0.125.

The service does not impose a maximum value. If you want to receive processing metrics only for transcription events instead of at periodic intervals, set the value to a large number. If the value is larger than the duration of the audio, the service returns processing metrics only for transcription events.

See [Processing metrics](#) .

Default: 1.0

audio_metrics If `true`, requests
bool detailed information about
the signal characteristics
of the input audio. The
service returns audio
metrics with the final
transcription results. By
default, the service returns
no audio metrics. See
[Audio metrics](#) .
Default: `false`

end_of_phrase_silence_time

float

Specifies the duration of
the pause interval at which
the service splits a
transcript into multiple
final results. If the service
detects pauses or
extended silence before it
reaches the end of the
audio stream, its response
can include multiple final
results. Silence indicates a
point at which the speaker
pauses between spoken
words or phrases.

Specify a value for the
pause interval in the range

of 0.0 to 120.0.

- A value greater than 0 specifies the interval that the service is to use for speech recognition.
- A value of 0 indicates that the service is to use the default interval. It is equivalent to omitting the parameter.

The default pause interval for most languages is 0.8 seconds. The default for Chinese is 0.6 seconds.

See [End of phrase silence time](#) .

Default: 0.8

split_transcript_at_phrase_end

bool

If `true` , directs the service to split the transcript into multiple final results based on semantic features of the input, for example, at the conclusion of meaningful phrases such as sentences.

The service bases its understanding of semantic features on the base language model that you use with a request. Custom language models and grammars can also influence how and where the service splits a transcript.

By default, the service splits transcripts based solely on the pause interval. If the parameters are used together on the same request,

`end_of_phrase_silence_time` has precedence over `split_transcript_at_phrase_end`.

See [Split transcript at phrase end](#).

Default: `false`

speech_detector_sensitivity

float

The sensitivity of speech activity detection that the service is to perform. Use

the parameter to suppress word insertions from music, coughing, and other non-speech events. The service biases the audio it passes for speech recognition by evaluating the input audio against prior models of speech and non-speech activity.

Specify a value between 0.0 and 1.0:

- 0.0 suppresses all audio (no speech is transcribed).
- 0.5 (the default) provides a reasonable compromise for the level of sensitivity.
- 1.0 suppresses no audio (speech detection sensitivity is disabled).

The values increase on a monotonic curve.

Specifying one or two decimal places of precision (for example, 0.55) is typically more than sufficient.

The parameter is supported with all large speech models, next-generation models and with most previous-generation models. See [Speech detector sensitivity](#) and [Language model support](#) .

Default: 0.5

background_audio_suppression

float

The level to which the service is to suppress background audio based on its volume to prevent it from being transcribed as speech. Use the parameter to suppress side conversations or background noise.

Specify a value between 0.0 and 1.0:

- 0.0 (the default) provides no suppression (background audio suppression is disabled).
- 0.5 provides a

reasonable level of audio suppression for general usage.

- 1.0 suppresses all audio (no audio is transcribed).

The values increase on a monotonic curve.

Specifying one or two decimal places of precision (for example, 0.55) is typically more than sufficient.

The parameter is supported with all large speech models, next-generation models and with most previous-generation models. See [Background audio suppression](#) and [Language model support](#).

Default: 0.0

low_latency	If true for next-generation Multimedia and Telephony models that support low latency, directs the service to
bool	

produce results even more quickly than it usually does. Next-generation models produce transcription results faster than previous-generation models. The `low_latency` parameter causes the models to produce results even more quickly, though the results might be less accurate when the parameter is used.

Note: The `low_latency` parameter is not available for large speech models and previous-generation `Broadband` and `Narrowband` models. It is available only for some next-generation models. To obtain interim results with a next-generation model, the model must support low latency and both the `interim_results` and `low_latency` parameters must be set to `true`.

- For a list of next-generation models that support low latency, see [Supported next-generation language models](#) .
- For more information about the `low_latency` parameter, see [Low latency](#) .

Default: `false`

For large speech models and next-generation models, an indication of whether the service is biased to recognize shorter or longer strings of characters when developing transcription hypotheses. By default, the service is optimized to produce the best balance of strings of different lengths.

The default bias is 0.0. The allowable range of values is -1.0 to 1.0.

- Negative values bias the

service to favor hypotheses with shorter strings of characters.

- Positive values bias the service to favor hypotheses with longer strings of characters.

As the value approaches -1.0 or 1.0, the impact of the parameter becomes more pronounced. To determine the most effective value for your scenario, start by setting the value of the parameter to a small increment, such as -0.1, -0.05, 0.05, or 0.1, and assess how the value impacts the transcription results. Then experiment with different values as necessary, adjusting the value by small increments.

Beta: The parameter is beta functionality. It is not available for previous-generation models.

See [Character insertion bias](#) .",

Response

Successful recognition returns one or more instances of a `SpeechRecognitionResults` object. The contents of the response depend on the parameters you send with the recognition request, including the `interim_results` parameter. For more information, see the results for the **Recognize audio** method.

Response handling

The `recognize_callback` parameter of the `recognize_using_websocket` method accepts an object of type `RecognizeCallback`. The object defines the methods that handle events from the WebSocket connection. You can override the definitions of the following default empty methods of the object to handle events that are associated with the connection and the request. The methods are called when their associated events occur.

Methods

`on_connected()`

The WebSocket connection is established.

Example response

```
{
  "results": [
    {
      "final": true,
      "alternatives": [
        {
          "transcript":
            "several tornadoes touch
            down as a line of severe
            thunderstorms swept
            through Colorado on Sunday
            ",
          "confidence":
            0.89
        },
        {
          "transcript":
            "several tornadoes touch
            down is a line of severe
            thunderstorms swept
            through Colorado on Sunday
            "
        },
        {
          "transcript":
            "several tornadoes touched
            down as a line of severe
            thunderstorms swept
            through Colorado on Sunday
            "
        }
      ]
    }
  ]
}
```

`on_listening()`

The service is listening for audio.

`on_data({data})`

Returns all response data for the request from the service.

`on_hypothesis({hypothesis})`

Returns interim results or maximum alternatives from the service when those responses are requested.

`on_transcription({transcript})`

Returns final transcription results for the request from the service.

`on_error({error})`

Reports an error in the WebSocket connection.

`on_inactivity_timeout({error})`

Reports an inactivity timeout for the request.

The connection can produce the following return codes.

Return code

```
],
  "keywords_result": {
    "tornadoes": [
      {
        "normalized_text":
        "tornadoes",
        "start_time":
        1.52,
        "end_time":
        2.15,
        "confidence":
        1.0
      }
    ],
    "colorado": [
      {
        "normalized_text":
        "Colorado",
        "start_time":
        4.95,
        "end_time":
        5.59,
        "confidence":
        0.98
      }
    ]
  },
  "result_index": 0
}
```

1000	The connection closed normally.
1002	The connection closed due to a protocol error.
1006	The connection closed abnormally.
1009	The connection closed because the frame size exceeded the 4 MB limit.
1011	The connection closed because the service encountered an unexpected internal condition that prevents it from fulfilling the request.

Methods

List models

Lists all language models that are available for use with the service. The information includes the name of the model and its minimum sampling rate in Hertz, among other things. The ordering of the list of models can change from call to call; do not rely on an alphabetized or static list of models.

See also: [Listing all models](#) .

Request

No Request Parameters


This method does not accept any request parameters.

```
list_models(  
    self,  
    **kwargs,  
) -> DetailedResponse
```



Example request for IBM Cloud





```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

speech_models =
speech_to_text.list_models
().get_result()
print(json.dumps(speech_mo
dels, indent=2))
```

Example request for IBM Cloud Pak for Data 

Response

SpeechModels

Information about the available language models.

Example responses

models

Always included *

An array of [SpeechModel](#) objects that provides information about each available model.

> [List\[SpeechModel\]](#)

Status Code

200

OK. The request succeeded.

406

Not Acceptable. The request specified an [Accept](#) header with an incompatible content type.

415

Unsupported Media Type. The request specified an unacceptable media type.

500

Internal Server Error. The service experienced an internal error.

503

Service Unavailable. The service is currently unavailable.

Success example

```
{
  "models": [
    {
      "name": "pt-
BR_NarrowbandModel",
      "language": "pt-BR",
      "url": "{url}/v1/models/pt-
BR_NarrowbandModel",
      "rate": 8000,

      "supported_features": {

        "custom_language_model":
true,

        "custom_acoustic_model":
true,
        "speaker_labels":
true
      },
      "description":
"Brazilian Portuguese
narrowband model."
    },
    {
      "name": "ko-
```

```
KR_BroadbandModel",
    "language": "ko-KR",
    "url": "{url}/models/ko-
KR_BroadbandModel",
    "rate": 16000,

    "supported_features": {

        "custom_language_model":
            true,

        "custom_acoustic_model":
            true,
            "speaker_labels":
            true
    },
    "description":
        "Korean broadband model."
    },
    {
        "name": "fr-
FR_BroadbandModel",
        "language": "fr-FR",
        "url": "{url}/v1/models/fr-
FR_BroadbandModel",
        "rate": 16000,

        "supported_features": {

            "custom_language_model":
                true,
```

```
"custom_acoustic_model":
true,
    "speaker_labels":
true
    },
    "description":
"French broadband model."
}
]
```

Get a model

Gets information for a single specified language model that is available for use with the service. The information includes the name of the model and its minimum sampling rate in Hertz, among other things.

See also: [Listing a specific model](#) .

Request

parameters

model_id	The identifier of the model in
-----------------	--------------------------------

```
get_model(
    self,
    model_id: str,
    **kwargs,
) -> DetailedResponse
```

Example request for IBM Cloud

Required *
str

the form of its name from the output of the [List models](#) method.

Allowable values: [

ar-MS_BroadbandModel ,
ar-MS_Telephony ,
cs-CZ_Telephony ,
de-DE_BroadbandModel ,
de-DE_Multimedia ,
de-DE_NarrowbandModel ,
de-DE_Telephony ,
en-AU_BroadbandModel ,
en-AU_Multimedia ,
en-AU_NarrowbandModel ,
en-AU_Telephony ,
en-GB_BroadbandModel ,
en-GB_Multimedia ,
en-GB_NarrowbandModel ,
en-GB_Telephony ,
en-IN_Telephony ,
en-US_BroadbandModel ,
en-US_Multimedia ,
en-US_NarrowbandModel ,
,
en-US_ShortForm_NarrowbandModel ,
en-US_Telephony ,

```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

speech_model =
speech_to_text.get_model('
en-
US_BroadbandModel').get_re
sult()
print(json.dumps(speech_mo
del, indent=2))
```

Example request for IBM Cloud Pak for Data

```
en-  
WW_Medical_Telephony  
, es-AR_BroadbandModel ,  
es-AR_NarrowbandModel  
, es-CL_BroadbandModel ,  
es-CL_NarrowbandModel  
, es-CO_BroadbandModel ,  
es-CO_NarrowbandModel  
, es-ES_BroadbandModel ,  
es-ES_NarrowbandModel  
, es-ES_Multimedia ,  
es-ES_Telephony ,  
es-LA_Telephony ,  
es-MX_BroadbandModel ,  
es-MX_NarrowbandModel  
, es-PE_BroadbandModel ,  
es-PE_NarrowbandModel  
, fr-CA_BroadbandModel ,  
fr-CA_Multimedia ,  
fr-CA_NarrowbandModel  
, fr-CA_Telephony ,  
fr-FR_BroadbandModel ,  
fr-FR_Multimedia ,  
fr-FR_NarrowbandModel  
, fr-FR_Telephony ,  
hi-IN_Telephony ,
```

```
it-IT_BroadbandModel ,
it-IT_NarrowbandModel
, it-IT_Multimedia ,
it-IT_Telephony ,
ja-JP_BroadbandModel ,
ja-JP_Multimedia ,
ja-JP_NarrowbandModel
, ja-JP_Telephony ,
ko-KR_BroadbandModel ,
ko-KR_Multimedia ,
ko-KR_NarrowbandModel
, ko-KR_Telephony ,
nl-BE_Telephony ,
nl-NL_BroadbandModel ,
nl-NL_Multimedia ,
nl-NL_NarrowbandModel
, nl-NL_Telephony ,
pt-BR_BroadbandModel ,
pt-BR_Multimedia ,
pt-BR_NarrowbandModel
, pt-BR_Telephony ,
sv-SE_Telephony ,
zh-CN_BroadbandModel ,
zh-CN_NarrowbandModel
, zh-CN_Telephony ]
```

Response

SpeechModel	Information about an available language model.
name Always included * str	The name of the model for use as an identifier in calls to the service (for example, <code>en-US_BroadbandModel</code>).
language Always included * str	The language identifier of the model (for example, <code>en-US</code>).
rate Always included * int	The sampling rate (minimum acceptable rate for audio) used by the model in Hertz.
url Always included * str	The URI for the model.
supported_features Always included * Indicates whether select service features are supported with the model.	
> SupportedFeatures	

Example responses

Success example

```
{
  "rate": 16000,
  "name": "en-US_BroadbandModel",
  "language": "en-US",
  "url": "{url}/v1/models/en-US_BroadbandModel",
  "supported_features": {
    "custom_language_model": true,
    "custom_acoustic_model": true,
    "speaker_labels": true
  },
  "description": "US English broadband model."
}
```

description A brief description of the
Always included * model.
str

Status Code

200 **OK.** The request succeeded.

404 **Not Found.** The specified
`model_id` was not found.

406 **Not Acceptable.** The request
specified an `Accept` header with
an incompatible content type.

415 **Unsupported Media Type.** The
request specified an unacceptable
media type.

500 **Internal Server Error.** The service
experienced an internal error.

503 **Service Unavailable.** The service is
currently unavailable.

Recognize audio

Sends audio and returns transcription results

`recognize(`

for a recognition request. You can pass a maximum of 100 MB and a minimum of 100 bytes of audio with a request. The service automatically detects the endianness of the incoming audio and, for audio that includes multiple channels, downmixes the audio to one-channel mono during transcoding. The method returns only final results; to enable interim results, use the WebSocket API. (With the `curl` command, use the `--data-binary` option to upload the file for the request.)

See also: [Making a basic HTTP request](#)

Streaming mode

For requests to transcribe live audio as it becomes available, you must set the `Transfer-Encoding` header to `chunked` to use streaming mode. In streaming mode, the service closes the connection (status code 408) if it does not receive at least 15 seconds of audio (including silence) in any 30-second period. The service also closes the connection (status code 400) if it detects no speech for `inactivity_timeout` seconds of streaming audio; use the `inactivity_timeout` parameter to change the default of 30 seconds.

See also:

- [Audio transmission](#)

```
self,
audio: BinaryIO,
*,
content_type: str
= None,
model: str = None,

language_customization_id:
str = None,

acoustic_customization_id:
str = None,

base_model_version: str =
None,

customization_weight:
float = None,

inactivity_timeout: int =
None,
keywords:
List[str] = None,

keywords_threshold: float
= None,
max_alternatives:
int = None,

word_alternatives_threshol
d: float = None,
word_confidence:
bool = None,
timestamps: bool =
```

- [Timeouts](#)

Audio formats (content types)

The service accepts audio in the following formats (MIME types).

- For formats that are labeled **Required**, you must use the `Content-Type` header with the request to specify the format of the audio.
- For all other formats, you can omit the `Content-Type` header or specify `application/octet-stream` with the header to have the service automatically detect the format of the audio. (With the `curl` command, you can specify either `"Content-Type:"` or `"Content-Type: application/octet-stream"` .)

Where indicated, the format that you specify must include the sampling rate and can optionally include the number of channels and the endianness of the audio.

- `audio/alaw` (**Required**. Specify the sampling rate (`rate`) of the audio.)
- `audio/basic` (**Required**. Use only with narrowband models.)
- `audio/flac`
- `audio/g729` (Use only with narrowband models.)

```
None,
    profanity_filter:
bool = None,
    smart_formatting:
bool = None,
    speaker_labels:
bool = None,
    grammar_name: str
= None,
    redaction: bool =
None,
    audio_metrics:
bool = None,

end_of_phrase_silence_time
: float = None,

split_transcript_at_phrase
_end: bool = None,

speech_detector_sensitivit
y: float = None,

background_audio_suppressi
on: float = None,
    low_latency: bool
= None,

character_insertion_bias:
float = None,
    **kwargs,
) -> DetailedResponse
```

- `audio/l16` (**Required.** Specify the sampling rate (`rate`) and optionally the number of channels (`channels`) and endianness (`endianness`) of the audio.)
- `audio/mp3`
- `audio/mpeg`
- `audio/mulaw` (**Required.** Specify the sampling rate (`rate`) of the audio.)
- `audio/ogg` (The service automatically detects the codec of the input audio.)
- `audio/ogg;codecs=opus`
- `audio/ogg;codecs=vorbis`
- `audio/wav` (Provide audio with a maximum of nine channels.)
- `audio/webm` (The service automatically detects the codec of the input audio.)
- `audio/webm;codecs=opus`
- `audio/webm;codecs=vorbis`

The sampling rate of the audio must match the sampling rate of the model for the recognition request: for broadband models, at least 16 kHz; for narrowband models, at least 8 kHz. If the sampling rate of the audio is higher than the minimum required rate, the service down-samples the audio to the appropriate rate. If the sampling rate of the audio is lower than the minimum required rate, the request fails.

See also: [Supported audio formats](#) .

Next-generation models

The service supports next-generation [Multimedia](#) (16 kHz) and [Telephony](#) (8 kHz) models for many languages. Next-generation models have higher throughput than the service's previous generation of [Broadband](#) and [Narrowband](#) models. When you use next-generation models, the service can return transcriptions more quickly and also provide noticeably better transcription accuracy.

You specify a next-generation model by using the [model](#) query parameter, as you do a previous-generation model. Most next-generation models support the [low_latency](#) parameter, and all next-generation models support the [character_insertion_bias](#) parameter. These parameters are not available with previous-generation models.

Next-generation models do not support all of the speech recognition parameters that are available for use with previous-generation models. Next-generation models do not support the following parameters:

- [acoustic_customization_id](#)
- [keywords](#) and [keywords_threshold](#)
- [processing_metrics](#) and [processing_metrics_interval](#)
- [word_alternatives_threshold](#)

Important: Effective **31 July 2023**, all previous-generation models will be removed from the service and the documentation. Most previous-generation models were deprecated on 15 March 2022. You must migrate to the equivalent next-generation model by 31 July 2023. For more information, see [Migrating to next-generation models](#) .

See also:

- [Next-generation languages and models](#)
- [Supported features for next-generation models](#)

Multipart speech recognition

Note: The asynchronous HTTP interface, WebSocket interface, and Watson SDKs do not support multipart speech recognition.

The HTTP **POST** method of the service also supports multipart speech recognition. With multipart requests, you pass all audio data as multipart form data. You specify some parameters as request headers and query parameters, but you pass JSON metadata as form data to control most aspects of the transcription. You can use multipart recognition to pass multiple audio files with a single request.

Use the multipart approach with browsers for

which JavaScript is disabled or when the parameters used with the request are greater than the 8 KB limit imposed by most HTTP servers and proxies. You can encounter this limit, for example, if you want to spot a very large number of keywords.

See also: [Making a multipart HTTP request](#) .

Request

parameters	
audio	The audio to transcribe.
Required *	
BinaryIO	

Example request for IBM Cloud

```
from os.path import join,
dirname
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
```

content_type
str

The format (MIME type) of the audio. For more information about specifying an audio format, see **Audio formats (content types)** in the method description.

Allowable values: [

```
application/octet-stream
, audio/alaw ,
audio/basic ,
audio/flac ,
audio/g729 , audio/l16 ,
audio/mp3 , audio/mpeg ,
audio/mulaw ,
audio/ogg ,
audio/ogg;codecs=opus
,
audio/ogg;codecs=vorbis
, audio/wav , audio/webm
,
audio/webm;codecs=opus
,
audio/webm;codecs=vorbis
]
```

model
str

The model to use for speech recognition. If you omit the

```
)

speech_to_text.set_service_url('{url}')

with
open(join(dirname(__file__), './.', 'audio-file2.flac'),
      'rb') as
audio_file:

speech_recognition_results
=
speech_to_text.recognize(
    audio=audio_file,
    content_type='audio/flac',

    word_alternatives_threshold=0.9,
    keywords=
['colorado', 'tornado',
'tornadoes'],

    keywords_threshold=0.5
).get_result()
print(json.dumps(speech_recognition_results,
    indent=2))
```

Download sample file [audio-file2.flac](#)

`model` parameter, the service uses the US English `en-US_BroadbandModel` by default.

For IBM Cloud Pak for Data, if you do not install the `en-US_BroadbandModel`, you must either specify a model with the request or specify a new default model for your installation of the service.

See also:

- [Using a model for speech recognition](#)
- [Using the default model](#)

Allowable values: [

`ar-MS_BroadbandModel` ,
`ar-MS_Telephony` ,
`cs-CZ_Telephony` ,
`de-DE_BroadbandModel` ,
`de-DE_Multimedia` ,
`de-DE_NarrowbandModel`
, `de-DE_Telephony` ,
`en-AU_BroadbandModel` ,
`en-AU_Multimedia` ,
`en-AU_NarrowbandModel`
, `en-AU_Telephony` ,

Example request for IBM Cloud Pak for Data 

en-IN_Telephony , en-GB_BroadbandModel , en-GB_Multimedia , en-GB_NarrowbandModel , en-GB_Telephony , en-US_BroadbandModel , en-US_Multimedia , en-US_NarrowbandModel ,	
en- US_ShortForm_NarrowbandModel , en-US_Telephony , en- WW_Medical_Telephony , es-AR_BroadbandModel , es-AR_NarrowbandModel , es-CL_BroadbandModel , es-CL_NarrowbandModel , es-CO_BroadbandModel , es-CO_NarrowbandModel , es-ES_BroadbandModel , es-ES_NarrowbandModel , es-ES_Multimedia , es-ES_Telephony , es-LA_Telephony , es-MX_BroadbandModel , es-MX_NarrowbandModel	

, es-PE_BroadbandModel ,
es-PE_NarrowbandModel
, fr-CA_BroadbandModel ,
fr-CA_Multimedia ,
fr-CA_NarrowbandModel
, fr-CA_Telephony ,
fr-FR_BroadbandModel ,
fr-FR_Multimedia ,
fr-FR_NarrowbandModel
, fr-FR_Telephony ,
hi-IN_Telephony ,
it-IT_BroadbandModel ,
it-IT_NarrowbandModel
, it-IT_Multimedia ,
it-IT_Telephony ,
ja-JP_BroadbandModel ,
ja-JP_Multimedia ,
ja-JP_NarrowbandModel
, ja-JP_Telephony ,
ko-KR_BroadbandModel ,
ko-KR_Multimedia ,
ko-KR_NarrowbandModel
, ko-KR_Telephony ,
nl-BE_Telephony ,
nl-NL_BroadbandModel ,
nl-NL_Multimedia ,

```
nl-NL_NarrowbandModel  
, nl-NL_Telephony ,  
pt-BR_BroadbandModel ,  
pt-BR_Multimedia ,  
pt-BR_NarrowbandModel  
, pt-BR_Telephony ,  
sv-SE_Telephony ,  
zh-CN_BroadbandModel ,  
zh-CN_NarrowbandModel  
, zh-CN_Telephony ]
```

Default: en-
US_BroadbandModel

language_customization_id

str

The customization ID (GUID) of a custom language model that is to be used with the recognition request. The base model of the specified custom language model must match the model specified with the `model` parameter. You must make the request with credentials for the instance of the service that owns the custom model. By default, no custom language model is used. See [Using a custom language model for speech recognition](#).

Note: Use this parameter instead of the deprecated `customization_id` parameter.

acoustic_customization_id

str

The customization ID (GUID) of a custom acoustic model that is to be used with the recognition request. The base model of the specified custom acoustic model must match the model specified with the `model` parameter. You must make the request with credentials for the instance of the service that owns the custom model. By default, no custom acoustic model is used. See [Using a custom acoustic model for speech recognition](#) .

base_model_version

str

The version of the specified base model that is to be used with the recognition request. Multiple versions of a base model can exist when a model is updated for internal improvements. The parameter is intended primarily for use with custom models that have been upgraded for a new base model. The default value depends on whether the parameter is used with or without a custom model. See [Making speech recognition requests with upgraded custom models](#) .

customization_weight

float

If you specify the customization ID (GUID) of a custom language model with the recognition request, the customization weight tells the service how much weight to give to words from the

custom language model compared to those from the base model for the current request.

Specify a value between 0.0 and 1.0. Unless a different customization weight was specified for the custom model when the model was trained, the default value is:

- 0.3 for previous-generation models
- 0.2 for most next-generation models
- 0.1 for next-generation English and Japanese models

A customization weight that you specify overrides a weight that was specified when the custom model was trained. The default value yields the best performance in general. Assign a higher value if your audio makes frequent use of OOV words from the custom model. Use caution when setting the weight: a higher value can improve the accuracy of

phrases from the custom model's domain, but it can negatively affect performance on non-domain phrases.

See [Using customization weight](#) .

inactivity_timeout

int

The time in seconds after which, if only silence (no speech) is detected in streaming audio, the connection is closed with a 400 error. The parameter is useful for stopping audio submission from a live microphone when a user simply walks away. Use `-1` for infinity. See [Inactivity timeout](#) .

Default: `30`

keywords

List[str]

An array of keyword strings to spot in the audio. Each keyword string can include one or more string tokens. Keywords are spotted only in the final results, not in interim hypotheses. If you specify any keywords, you must also specify a keywords threshold. Omit the parameter or specify an empty array if you do not need to spot keywords.

You can spot a maximum of 1000 keywords with a single request. A single keyword can have a maximum length of 1024 characters, though the maximum effective length for double-byte languages might be shorter. Keywords are case-insensitive.

See [Keyword spotting](#) .

keywords_threshold

float

A confidence value that is the lower bound for spotting a keyword. A word is considered to match a keyword if its confidence is greater than or equal to the threshold. Specify a probability between 0.0 and 1.0. If you specify a threshold, you must also specify one or more keywords. The service performs no keyword spotting if you omit either parameter. See [Keyword spotting](#) .

max_alternatives

int

The maximum number of alternative transcripts that the service is to return. By default, the service returns a single transcript. If you specify a value of 0, the service uses the default value, 1. See [Maximum alternatives](#).

Default: 1

word_alternatives_threshold

float

A confidence value that is the lower bound for identifying a hypothesis as a possible word alternative (also known as "Confusion Networks"). An alternative word is considered if its confidence is greater than or equal to the threshold. Specify a probability between 0.0 and 1.0. By default, the service computes no alternative words. See [Word alternatives](#).

word_confidence

bool

If **true** , the service returns a confidence measure in the range of 0.0 to 1.0 for each word. By default, the service returns no word confidence scores. See [Word confidence](#)

.

Default: **false**

timestamps

bool

If **true** , the service returns time alignment for each word. By default, no timestamps are returned. See [Word timestamps](#) .

Default: **false**

profanity_filter If `true` , the service filters profanity from all output except for keyword results by replacing inappropriate words with a series of asterisks. Set the parameter to `false` to return results with no censoring.

bool

Note: The parameter can be used with US English and Japanese transcription only. See [Profanity filtering](#) .

Default: `true`

smart_formatting

bool

If **true**, the service converts dates, times, series of digits and numbers, phone numbers, currency values, and internet addresses into more readable, conventional representations in the final transcript of a recognition request. For US English, the service also converts certain keyword strings to punctuation symbols. By default, the service performs no smart formatting.

Note: The parameter can be used with US English, Japanese, and Spanish (all dialects) transcription only.

See [Smart formatting](#).

Default: **false**

speaker_labels	If true , the response
bool	includes labels that identify which words were spoken by which participants in a multi-person exchange. By default, the service returns no speaker labels. Setting

`speaker_labels` to `true` forces the `timestamps` parameter to be `true`, regardless of whether you specify `false` for the parameter.

- *For previous-generation models*, the parameter can be used with Australian English, US English, German, Japanese, Korean, and Spanish (both broadband and narrowband models) and UK English (narrowband model) transcription only.
- *For next-generation models*, the parameter can be used with Czech, English (Australian, Indian, UK, and US), German, Japanese, Korean, and Spanish transcription only.

See [Speaker labels](#).

Default: `false`

grammar_name The name of a grammar that
str is to be used with the
recognition request. If you
specify a grammar, you must
also use the
`language_customization_id` parameter to specify
the name of the custom
language model for which
the grammar is defined. The
service recognizes only
strings that are recognized
by the specified grammar; it
does not recognize other
custom words from the
model's words resource.
See [Using a grammar for
speech recognition](#) .

redaction
bool If `true` , the service
redacts, or masks, numeric
data from final transcripts.
The feature redacts any
number that has three or
more consecutive digits by
replacing each digit with an
`X` character. It is intended
to redact sensitive numeric
data, such as credit card
numbers. By default, the

service performs no redaction.

When you enable redaction, the service automatically enables smart formatting, regardless of whether you explicitly disable that feature. To ensure maximum security, the service also disables keyword spotting (ignores the `keywords` and `keywords_threshold` parameters) and returns only a single final transcript (forces the `max_alternatives` parameter to be `1`).

Note: The parameter can be used with US English, Japanese, and Korean transcription only.

See [Numeric redaction](#) .

Default: `false`

audio_metrics If `true` , requests detailed information about the signal characteristics of the input audio. The service returns audio metrics with the final transcription results. By default, the service returns no audio metrics.

See [Audio metrics](#) .

Default: `false`

end_of_phrase_silence_time

float

Specifies the duration of the pause interval at which the service splits a transcript into multiple final results. If the service detects pauses or extended silence before it reaches the end of the audio stream, its response can include multiple final results.

Silence indicates a point at which the speaker pauses between spoken words or phrases.

Specify a value for the pause interval in the range of 0.0 to 120.0.

- A value greater than 0 specifies the interval that the service is to use for speech recognition.
- A value of 0 indicates that the service is to use the default interval. It is equivalent to omitting the parameter.

The default pause interval for most languages is 0.8 seconds; the default for Chinese is 0.6 seconds.

See [End of phrase silence time](#) .

Default: 0.8

split_transcript_at_phrase_end

bool

If **true** , directs the service to split the transcript into multiple final results based on semantic features of the input, for example, at the conclusion of meaningful phrases such as sentences.

The service bases its understanding of semantic features on the base language model that you use

with a request. Custom language models and grammars can also influence how and where the service splits a transcript.

By default, the service splits transcripts based solely on the pause interval. If the parameters are used together on the same request,

`end_of_phrase_silence_time` has precedence over `split_transcript_at_phrase_end`.

See [Split transcript at phrase end](#).

Default: `false`

speech_detector_sensitivity

float

The sensitivity of speech activity detection that the service is to perform. Use the parameter to suppress word insertions from music, coughing, and other non-speech events. The service biases the audio it passes for speech recognition by

evaluating the input audio against prior models of speech and non-speech activity.

Specify a value between 0.0 and 1.0:

- 0.0 suppresses all audio (no speech is transcribed).
- 0.5 (the default) provides a reasonable compromise for the level of sensitivity.
- 1.0 suppresses no audio (speech detection sensitivity is disabled).

The values increase on a monotonic curve. Specifying one or two decimal places of precision (for example, 0.55) is typically more than sufficient.

The parameter is supported with all next-generation models and with most previous-generation models.

See [Speech detector sensitivity](#) and [Language model support](#) .

Default: 0.5

background_audio_suppression

float

The level to which the service is to suppress background audio based on its volume to prevent it from being transcribed as speech.

Use the parameter to suppress side conversations or background noise.

Specify a value in the range of 0.0 to 1.0:

- 0.0 (the default) provides no suppression (background audio suppression is disabled).
- 0.5 provides a reasonable level of audio suppression for general usage.
- 1.0 suppresses all audio (no audio is transcribed).

The values increase on a monotonic curve. Specifying one or two decimal places of precision (for example, 0.55) is typically more than sufficient.

The parameter is supported with all next-generation

models and with most previous-generation models.

See [Background audio suppression](#) and [Language model support](#) .

Default: 0.0

low_latency

bool

If `true` for next-generation `Multimedia` and `Telephony` models that support low latency, directs the service to produce results even more quickly than it usually does. Next-generation models produce transcription results faster than previous-generation models. The `low_latency` parameter causes the models to produce results even more quickly, though the results might be less accurate when the parameter is used.

The parameter is not available for previous-generation `Broadband` and `Narrowband` models. It is available for most next-generation models.

- For a list of next-generation models that support low latency, see [Supported next-generation language models](#) .
- For more information about the `low_latency` parameter, see [Low latency](#) .

Default: `false`

character_insertion_bias

float

For next-generation models, an indication of whether the service is biased to recognize shorter or longer strings of characters when developing transcription hypotheses. By default, the service is optimized to produce the best balance of strings of different lengths.

The default bias is 0.0. The allowable range of values is -1.0 to 1.0.

- Negative values bias the service to favor hypotheses with shorter

strings of characters.

- Positive values bias the service to favor hypotheses with longer strings of characters.

As the value approaches -1.0 or 1.0, the impact of the parameter becomes more pronounced. To determine the most effective value for your scenario, start by setting the value of the parameter to a small increment, such as -0.1, -0.05, 0.05, or 0.1, and assess how the value impacts the transcription results. Then experiment with different values as necessary, adjusting the value by small increments.

The parameter is not available for previous-generation models.

See [Character insertion bias](#)

.

Default: 0.0

Response

SpeechRecognitionResultsThe complete results for a speech recognition request.

Example responses

Success example

```
{
  "results": [
    {
      "word_alternatives": [
        {
          "start_time": 0.15,
          "alternatives": [
            {
              "confidence": 1,
              "word": "a"
            }
          ],
          "end_time": 0.3
        },
        {
          "start_time": 0.3,
          "alternatives": [
            {
              "confidence": 1,
```

results

An array of

`SpeechRecognitionResult` objects that can include interim and final results (interim results are returned only if supported by the method). Final results are guaranteed not to change; interim results might be replaced by further interim results and eventually final results.

For the HTTP interfaces, all results arrive at the same time. For the WebSocket interface, results can be sent as multiple separate responses. The service periodically sends updates to the results list. The `result_index` is incremented to the lowest index in the array that has changed for new results.

For more information, see [Understanding speech recognition results](#).

> `List[SpeechRecognitionResult]`

```
        "word":
"line"
        }
    ],
    "end_time": 0.64
},
{
    "start_time":
0.64,
    "alternatives":
[
        {
"confidence": 1,
        "word": "of"
        }
    ],
    "end_time": 0.73
},
{
    "start_time":
0.73,
    "alternatives":
[
        {
"confidence": 1,
        "word":
"severe"
        }
    ],
    "end_time": 1.08
},
{
```

result_index

int

An index that indicates a change point in the `results` array. The service increments the index for additional results that it sends for new audio for the same request. All results with the same index are delivered at the same time. The same index can include multiple final results that are delivered with the same response.

```
        "start_time":
1.08,
        "alternatives":
[
    {

"confidence": 1,
        "word":
"thunderstorms"
    }
],
        "end_time": 1.85
    },
    {
        "start_time":
1.85,
        "alternatives":
[
    {

"confidence": 1,
        "word":
"with"
    }
],
        "end_time": 2
    },
    {
        "start_time": 2,
        "alternatives":
[
    {

"confidence": 1,
```

speaker_labels An array of `SpeakerLabelsResult` objects that identifies which words were spoken by which speakers in a multi-person exchange. The array is returned only if the `speaker_labels` parameter is `true`. When interim results are also requested for methods that support them, it is possible for a `SpeechRecognitionResults` object to include only the `speaker_labels` field.

> [List\[SpeakerLabelsResult\]](#)

processing_metrics

If processing metrics are requested, information about the service's processing of the input audio. Processing metrics are not available with the synchronous [Recognize audio](#) method.

> [ProcessingMetrics](#)

```
        "word":
"several"
        }
    ],
    "end_time": 2.52
},
{
    "start_time":
2.52,
    "alternatives":
[
    {
        "confidence": 1,
        "word":
"possible"
        }
    ],
    "end_time": 3.03
},
{
    "start_time":
3.03,
    "alternatives":
[
    {
        "confidence": 1,
        "word":
"tornadoes"
        }
    ],
    "end_time": 3.85
},
```

audio_metrics If audio metrics are requested, information about the signal characteristics of the input audio.

> [AudioMetrics](#)

warnings

List[str]

An array of warning messages associated with the request:

- Warnings for invalid parameters or fields can include a descriptive message and a list of invalid argument strings, for example, `"Unknown arguments:"` or `"Unknown url query arguments:"` followed by a list of the form `"{invalid_arg_1}, {invalid_arg_2}."` (If you use the `character_insertion_bias` parameter with a previous-generation model, the warning message refers to the parameter as `lambdaBias`.)
- The following warning is

```
{
  "start_time":
3.95,
  "alternatives":
[
  {
"confidence": 1,
    "word": "is"
  }
],
  "end_time": 4.13
},
{
  "start_time":
4.13,
  "alternatives":
[
  {
"confidence": 1,
    "word":
"approaching"
  }
],
  "end_time": 4.58
},
{
  "start_time":
4.58,
  "alternatives":
[
  {
```

returned if the request passes a custom model that is based on an older version of a base model for which an updated version is available:

"Using previous version of base model, because your custom model has been built with it. Please note that this version will be supported only for a limited time. Consider updating your custom model to the new base model. If you do not do that you will be automatically switched to base model when you used the non-updated custom model."

In both cases, the request succeeds despite the warnings.

```
"confidence": 0.96,
  "word":
    "Colorado"
  },
  "end_time": 5.16
},
{
  "start_time":
5.16,
  "alternatives":
[
  {
    "confidence": 0.95,
    "word": "on"
  },
  "end_time": 5.32
},
{
  "start_time":
5.32,
  "alternatives":
[
  {
    "confidence": 0.98,
    "word":
      "Sunday"
    },
    "end_time": 6.04
  }
}
```

Status Code

200 OK. The request succeeded.

400 **Bad Request.** The request failed because of a user input error. For example, the request passed audio that does not match the indicated format or failed to specify a required audio format; specified a custom language or custom acoustic model that is not in the [available](#) state; or experienced an inactivity timeout. Specific messages include

- `Model {model} not found`
- `Requested model is not available`
- `This 8000hz audio input requires a narrow band model. See /v1/models for a listp of available models.`
- `speaker_labels is not a supported feature for model {model}`
- `keywords_threshold` value must be between zero and one (inclusive)
- `word_alternatives_thresh`

```
],
  "keywords_result": {
    "colorado": [
      {
        "normalized_text":
        "Colorado",
        "start_time":
        4.58,
        "confidence":
        0.96,
        "end_time":
        5.16
      }
    ],
    "tornadoes": [
      {
        "normalized_text":
        "tornadoes",
        "start_time":
        3.03,
        "confidence":
        1,
        "end_time":
        3.85
      }
    ]
  },
  "alternatives": [
    {
      "confidence": 1,
      "transcript": "a
line of severe
```


old value must be between zero and one (inclusive)

- You cannot specify both 'customization_id' and 'language_customization_id' parameter!
- No speech detected for 30s
- Unable to transcode data stream application/octet-stream -> audio/l16
- Stream was {number} bytes but needs to be at least 100 bytes.
- keyword {keyword} length exceeds the maximum length 1024
- low_latency is not a supported feature for model {model}
- Character insertion bias must be a value between -1 and 1.

thunderstorms with several possible tornadoes is approaching Colorado on Sunday "

```
    }  
  ],  
  "final": true  
}  
],  
"result_index": 0  
}
```

404	Not Found. The specified model does not exist or, for IBM Cloud Pak for Data, the <code>model</code> parameter was not specified but the default model is not installed. The message is <code>Model '{model}' not found</code> .
406	Not Acceptable. The request specified an <code>Accept</code> header with an incompatible content type.
408	Request Timeout. The connection was closed due to inactivity (session timeout) for 30 seconds.
413	Payload Too Large. The request passed an audio file that exceeded the currently supported data limit.
415	Unsupported Media Type. The request specified an unacceptable media type.
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Register a callback

Registers a callback URL with the service for use with subsequent asynchronous recognition requests. The service attempts to register, or allowlist, the callback URL if it is not already registered by sending a `GET` request to the callback URL. The service passes a random alphanumeric challenge string via the `challenge_string` parameter of the request. The request includes an `Accept` header that specifies `text/plain` as the required response type.

To be registered successfully, the callback URL must respond to the `GET` request from the service. The response must send status code 200 and must include the challenge string in its body. Set the `Content-Type` response header to `text/plain`. Upon receiving this response, the service responds to the original registration request with response code 201.

The service sends only a single `GET` request to the callback URL. If the service does not receive a reply with a response code of 200 and a body that echoes the challenge string sent by the service within five seconds, it does not allowlist the URL; it instead sends status

```
register_callback(  
    self,  
    callback_url: str,  
    *,  
    user_secret: str =  
None,  
    **kwargs,  
) -> DetailedResponse
```



code 400 in response to the request to register a callback. If the requested callback URL is already allowlisted, the service responds to the initial registration request with response code 200.

If you specify a user secret with the request, the service uses it as a key to calculate an HMAC-SHA1 signature of the challenge string in its response to the `POST` request. It sends this signature in the `X-Callback-Signature` header of its `GET` request to the URL during registration. It also uses the secret to calculate a signature over the payload of every callback notification that uses the URL. The signature provides authentication and data integrity for HTTP communications.

After you successfully register a callback URL, you can use it with an indefinite number of recognition requests. You can register a maximum of 20 callback URLs in a one-hour span of time.

See also: [Registering a callback URL](#) .

Request

parameters

Example request for IBM Cloud



callback_url

Required *
str

An HTTP or HTTPS URL to which callback notifications are to be sent. To be allowlisted, the URL must successfully echo the challenge string during URL verification. During verification, the client can also check the signature that the service sends in the [X-Callback-Signature](#) header to verify the origin of the request.

user_secret

str

A user-specified string that the service uses to generate the HMAC-SHA1 signature that it sends via the [X-Callback-Signature](#) header. The service includes the header during URL verification and with every notification sent to the callback URL. It calculates the signature over the payload of the notification. If you omit the parameter, the service does not send the header.

```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

register_status =
speech_to_text.register_ca
llback(

'http://{user_callback_pat
h}/job_results',

user_secret='ThisIsMySecre
t'
).get_result()
print(json.dumps(register_
status, indent=2))
```

Response

RegisterStatus Information about a request to register a callback for asynchronous speech recognition.

status
Always included *
str

The current status of the job:

- created** : The service successfully allowlisted the callback URL as a result of the call.
- already created** : The URL was already allowlisted.

Possible values: [**created** , **already created**]

url
Always included *
str

The callback URL that is successfully registered.

Status Code

Example request for IBM Cloud Pak for Data

Example responses

Success example

```
{
  "status": "already
created",
  "url":
"http://{user_callback_pat
h}/job_results"
}
```

Success example

200	OK. The callback was already registered (allowlisted). The status included in the response is <code>already created</code> .
201	Created. The callback was successfully registered (allowlisted). The status included in the response is <code>created</code> .
400	Bad Request. The callback registration failed. The request was missing a required parameter or specified an invalid argument; the client sent an invalid response to the service's <code>GET</code> request during the registration process; or the client failed to respond to the server's request before the five-second timeout.
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Unregister a callback

Unregisters a callback URL that was previously allowlisted with a [Register a callback](#) request for use with the asynchronous interface. Once unregistered, the URL can no longer be used with asynchronous recognition requests.

See also: [Unregistering a callback URL](#) .

Request


parameters	
callback_url Required * str	The callback URL that is to be unregistered.

```
unregister_callback(  
    self,  
    callback_url: str,  
    **kwargs,  
) -> DetailedResponse
```



Example request for IBM Cloud





```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

speech_to_text.unregister_
callback('http://{user_cal
lback_path}/job_results')
```

Example request for IBM Cloud Pak for Data 

Response

Status Code	
200	OK. The callback URL was successfully unregistered.
400	Bad Request. The request failed because of a user input error (for example, because it failed to pass a callback URL).
404	Not Found. The specified callback URL was not found.
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Create a job

Creates a job for a new asynchronous recognition request. The job is owned by the instance of the service whose credentials are used to create it. How you learn the status and

```
create_job(  
    self,  
    audio: BinaryIO,  
    *,
```



results of a job depends on the parameters you include with the job creation request:

- By callback notification: Include the `callback_url` parameter to specify a URL to which the service is to send callback notifications when the status of the job changes. Optionally, you can also include the `events` and `user_token` parameters to subscribe to specific events and to specify a string that is to be included with each notification for the job.
- By polling the service: Omit the `callback_url`, `events`, and `user_token` parameters. You must then use the [Check jobs](#) or [Check a job](#) methods to check the status of the job, using the latter to retrieve the results when the job is complete.

The two approaches are not mutually exclusive. You can poll the service for job status or obtain results from the service manually even if you include a callback URL. In both cases, you can include the `results_ttl` parameter to specify how long the results are to remain available after the job is complete. Using the HTTPS [Check a job](#) method to retrieve results is more secure than receiving them via callback notification over HTTP because it provides confidentiality in addition to authentication and data integrity.

The method supports the same basic

```
content_type: str
= None,
model: str = None,
callback_url: str
= None,
events: str =
None,
user_token: str =
None,
results_ttl: int =
None,

language_customization_id:
str = None,

acoustic_customization_id:
str = None,

base_model_version: str =
None,

customization_weight:
float = None,

inactivity_timeout: int =
None,
keywords:
List[str] = None,

keywords_threshold: float
= None,
max_alternatives:
int = None,
```

parameters as other HTTP and WebSocket recognition requests. It also supports the following parameters specific to the asynchronous interface:

- `callback_url`
- `events`
- `user_token`
- `results_ttl`

You can pass a maximum of 1 GB and a minimum of 100 bytes of audio with a request. The service automatically detects the endianness of the incoming audio and, for audio that includes multiple channels, downmixes the audio to one-channel mono during transcoding. The method returns only final results; to enable interim results, use the WebSocket API. (With the `curl` command, use the `--data-binary` option to upload the file for the request.)

See also: [Creating a job](#) .

Streaming mode

For requests to transcribe live audio as it becomes available, you must set the `Transfer-Encoding` header to `chunked` to use streaming mode. In streaming mode, the service closes the connection (status code 408) if it does not receive at least 15 seconds

```
word_alternatives_threshold: float = None,
    word_confidence:
bool = None,
    timestamps: bool =
None,
    profanity_filter:
bool = None,
    smart_formatting:
bool = None,
    speaker_labels:
bool = None,
    grammar_name: str
= None,
    redaction: bool =
None,

processing_metrics: bool =
None,

processing_metrics_interval: float = None,
    audio_metrics:
bool = None,

end_of_phrase_silence_time
: float = None,

split_transcript_at_phrase
_end: bool = None,

speech_detector_sensitivity: float = None,
```

of audio (including silence) in any 30-second period. The service also closes the connection (status code 400) if it detects no speech for `inactivity_timeout` seconds of streaming audio; use the `inactivity_timeout` parameter to change the default of 30 seconds.

See also:

- [Audio transmission](#)
- [Timeouts](#)

Audio formats (content types)

The service accepts audio in the following formats (MIME types).

- For formats that are labeled **Required**, you must use the `Content-Type` header with the request to specify the format of the audio.
- For all other formats, you can omit the `Content-Type` header or specify `application/octet-stream` with the header to have the service automatically detect the format of the audio. (With the `curl` command, you can specify either `"Content-Type:"` or `"Content-Type: application/octet-stream"` .)

Where indicated, the format that you specify must include the sampling rate and can

```
background_audio_suppressi
on: float = None,
    low_latency: bool
= None,

character_insertion_bias:
float = None,
    **kwargs,
) -> DetailedResponse
```

optionally include the number of channels and the endianness of the audio.

- `audio/alaw` (**Required.** Specify the sampling rate (`rate`) of the audio.)
- `audio/basic` (**Required.** Use only with narrowband models.)
- `audio/flac`
- `audio/g729` (Use only with narrowband models.)
- `audio/l16` (**Required.** Specify the sampling rate (`rate`) and optionally the number of channels (`channels`) and endianness (`endianness`) of the audio.)
- `audio/mp3`
- `audio/mpeg`
- `audio/mulaw` (**Required.** Specify the sampling rate (`rate`) of the audio.)
- `audio/ogg` (The service automatically detects the codec of the input audio.)
- `audio/ogg;codecs=opus`
- `audio/ogg;codecs=vorbis`
- `audio/wav` (Provide audio with a maximum of nine channels.)
- `audio/webm` (The service automatically detects the codec of the input audio.)
- `audio/webm;codecs=opus`
- `audio/webm;codecs=vorbis`

The sampling rate of the audio must match the sampling rate of the model for the recognition request: for broadband models, at least 16 kHz; for narrowband models, at least 8 kHz. If the sampling rate of the audio is higher than the minimum required rate, the service down-samples the audio to the appropriate rate. If the sampling rate of the audio is lower than the minimum required rate, the request fails.

See also: [Supported audio formats](#) .

Next-generation models

The service supports next-generation [Multimedia](#) (16 kHz) and [Telephony](#) (8 kHz) models for many languages. Next-generation models have higher throughput than the service's previous generation of [Broadband](#) and [Narrowband](#) models. When you use next-generation models, the service can return transcriptions more quickly and also provide noticeably better transcription accuracy.

You specify a next-generation model by using the [model](#) query parameter, as you do a previous-generation model. Most next-generation models support the [low_latency](#) parameter, and all next-generation models support the [character_insertion_bias](#) parameter. These parameters are not available with previous-generation models.

Next-generation models do not support all of the speech recognition parameters that are available for use with previous-generation models. Next-generation models do not support the following parameters:

- `acoustic_customization_id`
- `keywords` and `keywords_threshold`
- `processing_metrics` and `processing_metrics_interval`
- `word_alternatives_threshold`

Important: Effective **31 July 2023**, all previous-generation models will be removed from the service and the documentation. Most previous-generation models were deprecated on 15 March 2022. You must migrate to the equivalent next-generation model by 31 July 2023. For more information, see [Migrating to next-generation models](#) .

See also:

- [Next-generation languages and models](#)
- [Supported features for next-generation models](#) .

Request

parameters

Example request for IBM Cloud



audio

The audio to transcribe.

Required *

BinaryIO

```
from os.path import join,
dirname
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

with
open(join(dirname(__file__
), './.', 'audio-
file.flac'),
      'rb') as
audio_file:
    recognition_job =
speech_to_text.create_job(
    audio_file,
```

content_type

str

The format (MIME type) of the audio. For more information about specifying an audio format, see **Audio formats (content types)** in the method description.

Allowable values: [

```
application/octet-stream
, audio/alaw ,
audio/basic ,
audio/flac ,
audio/g729 , audio/l16 ,
audio/mp3 , audio/mpeg ,
audio/mulaw ,
audio/ogg ,
audio/ogg;codecs=opus
,
audio/ogg;codecs=vorbis
, audio/wav , audio/webm
,
audio/webm;codecs=opus
,
audio/webm;codecs=vorbis
]
```

model

str

The model to use for speech recognition. If you omit the

```
content_type='audio/flac',
```

```
callback_url='http://{user_
_callback_path}/job_result
s',
```

```
user_token='job25',
        timestamps=True
).get_result()
print(json.dumps(recogniti
on_job, indent=2))
```

Download sample file [audio-file.flac](#)

Example request for IBM Cloud Pak for Data 

`model` parameter, the service uses the US English `en-US_BroadbandModel` by default.

For IBM Cloud Pak for Data, if you do not install the `en-US_BroadbandModel`, you must either specify a model with the request or specify a new default model for your installation of the service.

See also:

- [Using a model for speech recognition](#)
- [Using the default model](#)

Allowable values: [

`ar-MS_BroadbandModel` ,
`ar-MS_Telephony` ,
`cs-CZ_Telephony` ,
`de-DE_BroadbandModel` ,
`de-DE_Multimedia` ,
`de-DE_NarrowbandModel`
, `de-DE_Telephony` ,
`en-AU_BroadbandModel` ,
`en-AU_Multimedia` ,
`en-AU_NarrowbandModel`
, `en-AU_Telephony` ,

en-IN_Telephony , en-GB_BroadbandModel , en-GB_Multimedia , en-GB_NarrowbandModel , en-GB_Telephony , en-US_BroadbandModel , en-US_Multimedia , en-US_NarrowbandModel ,	
en- US_ShortForm_NarrowbandModel , en-US_Telephony , en- WW_Medical_Telephony , es-AR_BroadbandModel , es-AR_NarrowbandModel , es-CL_BroadbandModel , es-CL_NarrowbandModel , es-CO_BroadbandModel , es-CO_NarrowbandModel , es-ES_BroadbandModel , es-ES_NarrowbandModel , es-ES_Multimedia , es-ES_Telephony , es-LA_Telephony , es-MX_BroadbandModel , es-MX_NarrowbandModel	

, es-PE_BroadbandModel ,
es-PE_NarrowbandModel
, fr-CA_BroadbandModel ,
fr-CA_Multimedia ,
fr-CA_NarrowbandModel
, fr-CA_Telephony ,
fr-FR_BroadbandModel ,
fr-FR_Multimedia ,
fr-FR_NarrowbandModel
, fr-FR_Telephony ,
hi-IN_Telephony ,
it-IT_BroadbandModel ,
it-IT_NarrowbandModel
, it-IT_Multimedia ,
it-IT_Telephony ,
ja-JP_BroadbandModel ,
ja-JP_Multimedia ,
ja-JP_NarrowbandModel
, ja-JP_Telephony ,
ko-KR_BroadbandModel ,
ko-KR_Multimedia ,
ko-KR_NarrowbandModel
, ko-KR_Telephony ,
nl-BE_Telephony ,
nl-NL_BroadbandModel ,
nl-NL_Multimedia ,

```
nl-NL_NarrowbandModel  
, nl-NL_Telephony ,  
pt-BR_BroadbandModel ,  
pt-BR_Multimedia ,  
pt-BR_NarrowbandModel  
, pt-BR_Telephony ,  
sv-SE_Telephony ,  
zh-CN_BroadbandModel ,  
zh-CN_NarrowbandModel  
, zh-CN_Telephony ]
```

Default: en-
US_BroadbandModel

callback_url

str

A URL to which callback notifications are to be sent. The URL must already be successfully allowlisted by using the [Register a callback](#) method. You can include the same callback URL with any number of job creation requests. Omit the parameter to poll the service for job completion and results.

Use the `user_token` parameter to specify a unique user-specified string with each job to differentiate the callback notifications for the jobs.

events

str

If the job includes a callback URL, a comma-separated list of notification events to which to subscribe. Valid events are

- `recognitions.started` generates a callback notification when the service begins to process the job.
- `recognitions.comple`

`ted` generates a callback notification when the job is complete. You must use the [Check a job](#) method to retrieve the results before they time out or are deleted.

- `recognitions.completed_with_results` generates a callback notification when the job is complete. The notification includes the results of the request.
- `recognitions.failed` generates a callback notification if the service experiences an error while processing the job.

The

`recognitions.completed` and

`recognitions.completed_with_results` events

are incompatible. You can specify only one of the two events.

If the job includes a callback URL, omit the parameter to subscribe to the default

events:

`recognitions.started` ,
`recognitions.completed`
`d` , and
`recognitions.failed` .

If the job does not include a
callback URL, omit the
parameter.

Allowable values: [

`recognitions.started` ,
`recognitions.completed`
,
`recognitions.completed_with_results`
, `recognitions.failed`]

user_token
str

If the job includes a callback
URL, a user-specified string
that the service is to include
with each callback
notification for the job; the
token allows the user to
maintain an internal mapping
between jobs and
notification events. If the job
does not include a callback
URL, omit the parameter.

results_ttl

int

The number of minutes for which the results are to be available after the job has finished. If not delivered via a callback, the results must be retrieved within this time. Omit the parameter to use a time to live of one week. The parameter is valid with or without a callback URL.

language_customization_id

str

The customization ID (GUID) of a custom language model that is to be used with the recognition request. The base model of the specified custom language model must match the model specified with the `model` parameter. You must make the request with credentials for the instance of the service that owns the custom model. By default, no custom language model is used. See [Using a custom language model for speech recognition](#).

Note: Use this parameter instead of the deprecated `customization_id` parameter.

acoustic_customization_id

str

The customization ID (GUID) of a custom acoustic model that is to be used with the recognition request. The base model of the specified custom acoustic model must match the model specified with the `model` parameter. You must make the request with credentials for the instance of the service that owns the custom model. By default, no custom acoustic model is used. See [Using a custom acoustic model for speech recognition](#) .

base_model_version

str

The version of the specified base model that is to be used with the recognition request. Multiple versions of a base model can exist when a model is updated for internal improvements. The parameter is intended primarily for use with custom models that have been upgraded for a new base model. The default value depends on whether the parameter is used with or without a custom model. See [Making speech recognition requests with upgraded custom models](#) .

customization_weight

float

If you specify the customization ID (GUID) of a custom language model with the recognition request, the customization weight tells the service how much weight to give to words from the

custom language model compared to those from the base model for the current request.

Specify a value between 0.0 and 1.0. Unless a different customization weight was specified for the custom model when the model was trained, the default value is:

- 0.3 for previous-generation models
- 0.2 for most next-generation models
- 0.1 for next-generation English and Japanese models

A customization weight that you specify overrides a weight that was specified when the custom model was trained. The default value yields the best performance in general. Assign a higher value if your audio makes frequent use of OOV words from the custom model. Use caution when setting the weight: a higher value can improve the accuracy of

phrases from the custom model's domain, but it can negatively affect performance on non-domain phrases.

See [Using customization weight](#) .

inactivity_timeout

int

The time in seconds after which, if only silence (no speech) is detected in streaming audio, the connection is closed with a 400 error. The parameter is useful for stopping audio submission from a live microphone when a user simply walks away. Use `-1` for infinity. See [Inactivity timeout](#) .

Default: `30`

keywords

List[str]

An array of keyword strings to spot in the audio. Each keyword string can include one or more string tokens. Keywords are spotted only in the final results, not in interim hypotheses. If you specify any keywords, you must also specify a keywords threshold. Omit the parameter or specify an empty array if you do not need to spot keywords.

You can spot a maximum of 1000 keywords with a single request. A single keyword can have a maximum length of 1024 characters, though the maximum effective length for double-byte languages might be shorter. Keywords are case-insensitive.

See [Keyword spotting](#) .

keywords_threshold

float

A confidence value that is the lower bound for spotting a keyword. A word is considered to match a keyword if its confidence is greater than or equal to the threshold. Specify a probability between 0.0 and 1.0. If you specify a threshold, you must also specify one or more keywords. The service performs no keyword spotting if you omit either parameter. See [Keyword spotting](#) .

max_alternatives

int

The maximum number of alternative transcripts that the service is to return. By default, the service returns a single transcript. If you specify a value of **0** , the service uses the default value, **1** . See [Maximum alternatives](#) .

Default: **1**

word_alternatives_threshold

float

A confidence value that is the lower bound for identifying a hypothesis as a possible word alternative (also known as "Confusion Networks"). An alternative word is considered if its confidence is greater than or equal to the threshold. Specify a probability between 0.0 and 1.0. By default, the service computes no alternative words. See [Word alternatives](#) .

word_confidence

bool

If **true** , the service returns a confidence measure in the range of 0.0 to 1.0 for each word. By default, the service returns no word confidence scores. See [Word confidence](#)

.

Default: **false**

timestamps

bool

If **true** , the service returns time alignment for each word. By default, no timestamps are returned. See [Word timestamps](#) .

Default: **false**

profanity_filter If `true` , the service filters profanity from all output except for keyword results by replacing inappropriate words with a series of asterisks. Set the parameter to `false` to return results with no censoring.

bool

Note: The parameter can be used with US English and Japanese transcription only. See [Profanity filtering](#) .

Default: `true`

smart_formatting

bool

If **true**, the service converts dates, times, series of digits and numbers, phone numbers, currency values, and internet addresses into more readable, conventional representations in the final transcript of a recognition request. For US English, the service also converts certain keyword strings to punctuation symbols. By default, the service performs no smart formatting.

Note: The parameter can be used with US English, Japanese, and Spanish (all dialects) transcription only.

See [Smart formatting](#) .

Default: **false**

speaker_labels	If true , the response
bool	includes labels that identify which words were spoken by which participants in a multi-person exchange. By default, the service returns no speaker labels. Setting

`speaker_labels` to `true` forces the `timestamps` parameter to be `true`, regardless of whether you specify `false` for the parameter.

- *For previous-generation models*, the parameter can be used with Australian English, US English, German, Japanese, Korean, and Spanish (both broadband and narrowband models) and UK English (narrowband model) transcription only.
- *For next-generation models*, the parameter can be used with Czech, English (Australian, Indian, UK, and US), German, Japanese, Korean, and Spanish transcription only.

See [Speaker labels](#).

Default: `false`

grammar_name str	The name of a grammar that is to be used with the recognition request. If you specify a grammar, you must also use the <code>language_customization_id</code> parameter to specify the name of the custom language model for which the grammar is defined. The service recognizes only strings that are recognized by the specified grammar; it does not recognize other custom words from the model's words resource. See Using a grammar for speech recognition .
----------------------------	--

redaction bool	If <code>true</code> , the service redacts, or masks, numeric data from final transcripts. The feature redacts any number that has three or more consecutive digits by replacing each digit with an <code>X</code> character. It is intended to redact sensitive numeric data, such as credit card numbers. By default, the
--------------------------	---

service performs no redaction.

When you enable redaction, the service automatically enables smart formatting, regardless of whether you explicitly disable that feature. To ensure maximum security, the service also disables keyword spotting (ignores the `keywords` and `keywords_threshold` parameters) and returns only a single final transcript (forces the `max_alternatives` parameter to be `1`).

Note: The parameter can be used with US English, Japanese, and Korean transcription only.

See [Numeric redaction](#) .

Default: `false`

processing_metrics

bool

If **true**, requests

processing metrics about the service's transcription of the input audio. The service returns processing metrics at the interval specified by the **processing_metrics_interval** parameter. It also returns processing metrics for transcription events, for example, for final and interim results. By default, the service returns no processing metrics.

See [Processing metrics](#) .

Default: **false**

processing_metrics_interval

float

Specifies the interval in real wall-clock seconds at which the service is to return processing metrics. The parameter is ignored unless the `processing_metrics` parameter is set to `true`.

The parameter accepts a minimum value of 0.1 seconds. The level of precision is not restricted, so you can specify values such as 0.25 and 0.125.

The service does not impose a maximum value. If you want to receive processing metrics only for transcription events instead of at periodic intervals, set the value to a large number. If the value is larger than the duration of the audio, the service returns processing metrics only for transcription events.

See [Processing metrics](#).

Default: `1.0`

audio_metrics If `true` , requests detailed information about the signal characteristics of the input audio. The service returns audio metrics with the final transcription results. By default, the service returns no audio metrics.

See [Audio metrics](#) .

Default: `false`

end_of_phrase_silence_time

float

Specifies the duration of the pause interval at which the service splits a transcript into multiple final results. If the service detects pauses or extended silence before it reaches the end of the audio stream, its response can include multiple final results. Silence indicates a point at which the speaker pauses between spoken words or phrases.

Specify a value for the pause interval in the range of 0.0 to 120.0.

- A value greater than 0 specifies the interval that the service is to use for speech recognition.
- A value of 0 indicates that the service is to use the default interval. It is equivalent to omitting the parameter.

The default pause interval for most languages is 0.8 seconds; the default for Chinese is 0.6 seconds.

See [End of phrase silence time](#) .

Default: 0.8

split_transcript_at_phrase_end

bool

If **true** , directs the service to split the transcript into multiple final results based on semantic features of the input, for example, at the conclusion of meaningful phrases such as sentences.

The service bases its understanding of semantic features on the base language model that you use

with a request. Custom language models and grammars can also influence how and where the service splits a transcript.

By default, the service splits transcripts based solely on the pause interval. If the parameters are used together on the same request,

`end_of_phrase_silence_time` has precedence over `split_transcript_at_phrase_end`.

See [Split transcript at phrase end](#).

Default: `false`

speech_detector_sensitivity

float

The sensitivity of speech activity detection that the service is to perform. Use the parameter to suppress word insertions from music, coughing, and other non-speech events. The service biases the audio it passes for speech recognition by

evaluating the input audio against prior models of speech and non-speech activity.

Specify a value between 0.0 and 1.0:

- 0.0 suppresses all audio (no speech is transcribed).
- 0.5 (the default) provides a reasonable compromise for the level of sensitivity.
- 1.0 suppresses no audio (speech detection sensitivity is disabled).

The values increase on a monotonic curve. Specifying one or two decimal places of precision (for example, 0.55) is typically more than sufficient.

The parameter is supported with all next-generation models and with most previous-generation models.

See [Speech detector sensitivity](#) and [Language model support](#) .

Default: 0.5

background_audio_suppression

float

The level to which the service is to suppress background audio based on its volume to prevent it from being transcribed as speech.

Use the parameter to suppress side conversations or background noise.

Specify a value in the range of 0.0 to 1.0:

- 0.0 (the default) provides no suppression (background audio suppression is disabled).
- 0.5 provides a reasonable level of audio suppression for general usage.
- 1.0 suppresses all audio (no audio is transcribed).

The values increase on a monotonic curve. Specifying one or two decimal places of precision (for example, 0.55) is typically more than sufficient.

The parameter is supported with all next-generation

models and with most previous-generation models.

See [Background audio suppression](#) and [Language model support](#) .

Default: 0.0

low_latency

bool

If `true` for next-generation `Multimedia` and `Telephony` models that support low latency, directs the service to produce results even more quickly than it usually does. Next-generation models produce transcription results faster than previous-generation models. The `low_latency` parameter causes the models to produce results even more quickly, though the results might be less accurate when the parameter is used.

The parameter is not available for previous-generation `Broadband` and `Narrowband` models. It is available for most next-generation models.

- For a list of next-generation models that support low latency, see [Supported next-generation language models](#) .
- For more information about the `low_latency` parameter, see [Low latency](#) .

Default: `false`

character_insertion_bias

float

For next-generation models, an indication of whether the service is biased to recognize shorter or longer strings of characters when developing transcription hypotheses. By default, the service is optimized to produce the best balance of strings of different lengths.

The default bias is 0.0. The allowable range of values is -1.0 to 1.0.

- Negative values bias the service to favor hypotheses with shorter

strings of characters.

- Positive values bias the service to favor hypotheses with longer strings of characters.

As the value approaches -1.0 or 1.0, the impact of the parameter becomes more pronounced. To determine the most effective value for your scenario, start by setting the value of the parameter to a small increment, such as -0.1, -0.05, 0.05, or 0.1, and assess how the value impacts the transcription results. Then experiment with different values as necessary, adjusting the value by small increments.

The parameter is not available for previous-generation models.

See [Character insertion bias](#)

.

Default: 0.0

Response

RecognitionJob Information about a current asynchronous speech recognition job.

id The ID of the asynchronous job.
Always included *
str

status The current status of the job:
Always included *
str

- **waiting** : The service is preparing the job for processing. The service returns this status when the job is initially created or when it is waiting for capacity to process the job. The job remains in this state until the service has the capacity to begin processing it.
- **processing** : The service is actively processing the job.
- **completed** : The service has finished processing the job. If the job specified a callback URL

Example responses

Success example

```
{
  "id": "4bd734c0-e575-21f3-de03-f932aa0468a0",
  "status": "waiting",
  "created": "2016-08-17T19:15:17.926Z",
  "url": "{url}/v1/recognitions/4bd734c0-e575-21f3-de03-f932aa0468a0"
}
```

and the event

`recognitions.comple
ted_with_results` ,

the service sent the
results with the callback
notification. Otherwise,
you must retrieve the
results by checking the
individual job.

- `failed` : The job failed.

Possible values: [`waiting` ,
`processing` , `completed` ,
`failed`]

created

Always included *
str

The date and time in
Coordinated Universal Time
(UTC) at which the job was
created. The value is
provided in full ISO 8601
format (`YYYY-MM-
DDThh:mm:ss.sTZD`).

updated

str

The date and time in Coordinated Universal Time (UTC) at which the job was last updated by the service. The value is provided in full ISO 8601 format (`YYYY-MM-DDThh:mm:ss.sTZD`). This field is returned only by the [Check jobs](#) and [Check a job](#checkjob) methods.

url

str

The URL to use to request information about the job with the [Check a job](#) method. This field is returned only by the [Create a job](#) method.

user_token

str

The user token associated with a job that was created with a callback URL and a user token. This field can be returned only by the [Check jobs](#) method.

results

If the status is `completed`, the results of the recognition request as an array that includes a single instance of a `SpeechRecognitionResults` object. This field is returned only by the [Check a job](#) method.

> [List\[SpeechRecognitionResults\]](#)

warnings

List[str]

An array of warning messages about invalid parameters included with the request. Each warning includes a descriptive message and a list of invalid argument strings, for example, "unexpected query parameter 'user_token', query parameter 'callback_url' was not specified" . The request succeeds despite the warnings. This field can be returned only by the [Create a job](#) method. (If you use the `character_insertion_bias` parameter with a previous-generation model, the warning message refers to the parameter as `lambdaBias` .).

Status Code

201

Created. The job was successfully created.

400

Bad Request. The request failed

because of a user input error. For example, the request passed audio that does not match the indicated format or failed to specify a required audio format; specified a custom language or custom acoustic model that is not in the `available` state; or specified both the `recognitions.completed` and `recognitions.completed_with_results` events. Specific messages include

- `Model {model} not found`
- `Requested model is not available`
- `This 8000hz audio input requires a narrow band model. See /v1/models for a list of available models.`
- `speaker_labels is not a supported feature for model {model}`
- `keywords_threshold` value must be between zero and one (inclusive)
- `word_alternatives_threshold` value must be between zero and one (inclusive)

- You cannot specify both 'customization_id' and 'language_customization_id' parameter!
- No speech detected for 30s
- Unable to transcode data stream application/octet-stream -> audio/l16
- Stream was {number} bytes but needs to be at least 100 bytes.
- keyword {keyword} length exceeds the maximum length 1024
- low_latency is not a supported feature for model {model}
- Character insertion bias must be a value between -1 and 1.

404

Not Found. The specified model does not exist or, for IBM Cloud Pak for Data, the `model` parameter was not specified but the default model is not installed. The message is `Model '{model}' not found`.

500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Check jobs

Returns the ID and status of the latest 100 outstanding jobs associated with the credentials with which it is called. The method also returns the creation and update times of each job, and, if a job was created with a callback URL and a user token, the user token for the job. To obtain the results for a job whose status is **completed** or not one of the latest 100 outstanding jobs, use the [Check a job](#checkjob) method. A job and its results remain available until you delete them with the [Delete a job](#) method or until the job's time to live expires, whichever comes first.

See also: [Checking the status of the latest jobs](#)

Request


No Request Parameters

```
check_jobs(  
    self,  
    **kwargs,  
) -> DetailedResponse
```



This method does not accept any request parameters.

Example request for IBM Cloud 



```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

recognition_jobs =
speech_to_text.check_jobs(
).get_result()
print(json.dumps(recogniti
on_jobs, indent=2))
```

Example request for IBM Cloud Pak for Data 

Response

RecognitionJobs Information about current asynchronous speech recognition jobs.

recognitions An array of **RecognitionJob** objects that provides the status for each of the user's current jobs. The array is empty if the user has no current jobs.

Always included *

> [List\[RecognitionJob\]](#)

Status Code

200 OK. The request succeeded.

500 **Internal Server Error.** The service experienced an internal error.

503 **Service Unavailable.** The service is currently unavailable.

Example responses

Success example

```
{
  "recognitions": [
    {
      "id": "4bd734c0-e575-21f3-de03-f932aa0468a0",
      "created": "2016-08-17T19:15:17.926Z",
      "updated": "2016-08-17T19:15:17.926Z",
      "status": "waiting",
      "user_token": "job25"
    },
    {
      "id": "4bb1dca0-f6b1-11e5-80bc-71fb7b058b20",
      "created": "2016-08-17T19:13:23.622Z",
      "updated": "2016-08-17T19:13:24.434Z",
      "status": "processing"
    }
  ]
}
```

```
      "id": "398fcd80-  
330a-22ba-93ce-  
1a73f454dd98",  
      "created": "2016-08-  
17T19:11:04.298Z",  
      "updated": "2016-08-  
17T19:11:16.003Z",  
      "status":  
      "completed"  
    }  
  ]  
}
```

Check a job

Returns information about the specified job. The response always includes the status of the job and its creation and update times. If the status is `completed`, the response includes the results of the recognition request. You must use credentials for the instance of the service that owns a job to list information about it.

You can use the method to retrieve the results of any job, regardless of whether it was submitted with a callback URL and the `recognitions.completed_with_results` event, and you can retrieve the results multiple times for as long as they remain available. Use the [Check jobs](#) method to request information about the most recent jobs associated with the calling credentials.

See also: [Checking the status and retrieving the results of a job](#) .

Request

parameters

```
check_job(  
    self,  
    id: str,  
    **kwargs,  
) -> DetailedResponse
```



Example request for IBM Cloud



id

Required *
str

The identifier of the asynchronous job that is to be used for the request. You must make the request with credentials for the instance of the service that owns the job.

```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

recognition_job =
speech_to_text.check_job({
id}).get_result()
print(json.dumps(recogniti
on_job, indent=2))
```

Example request for IBM Cloud Pak for Data 

Response

RecognitionJob Information about a current asynchronous speech recognition job.

Example responses

Success example

```
{
  "created": "2016-08-17T19:11:04.298Z",
  "id": "4bd734c0-e575-21f3-de03-f932aa0468a0",
  "updated": "2016-08-17T19:11:16.003Z",
  "results": [
    {
      "result_index": 0,
      "results": [
        {
          "final": true,
          "alternatives": [
            {
              "transcript": "several tornadoes touch down as a line of severe thunderstorms swept through Colorado on Sunday"
            }
          ]
        }
      ]
    }
  ],
  "timestamps": [
    [

```

id
Always included *
str
The ID of the asynchronous job.

status
Always included *
str
The current status of the job:

- waiting** : The service is preparing the job for processing. The service returns this status when the job is initially created or when it is waiting for capacity to process the job. The job remains in this state until the service has the capacity to begin processing it.
- processing** : The service is actively processing the job.
- completed** : The service has finished processing the job. If the job specified a callback URL and the event `recognitions.comple`

ted_with_results ,
the service sent the
results with the callback
notification. Otherwise,
you must retrieve the
results by checking the
individual job.

- failed : The job failed.

Possible values: [waiting ,
processing , completed ,
failed]

created

Always included *
str

The date and time in
Coordinated Universal Time
(UTC) at which the job was
created. The value is
provided in full ISO 8601
format (YYYY-MM-
DDThh:mm:ss.sTZD).

```
"several" ,  
    1,  
    1.52  
],  
[  
  
"tornadoes" ,  
    1.52 ,  
    2.15  
],  
[  
    "touch" ,  
    2.15 ,  
    2.49  
],  
[  
    "down" ,  
    2.49 ,  
    2.82  
],  
[  
    "as" ,  
    2.82 ,  
    2.92  
],  
[  
    "a" ,  
    2.92 ,  
    3.01  
],  
[  
    "line" ,  
    3.01 ,  
    3.3
```

updated str	The date and time in Coordinated Universal Time (UTC) at which the job was last updated by the service. The value is provided in full ISO 8601 format (YYYY-MM-DDThh:mm:ss.sTZD). This field is returned only by the Check jobs and [Check a job[(#checkjob) methods.
url str	The URL to use to request information about the job with the Check a job method. This field is returned only by the Create a job method.
user_token str	The user token associated with a job that was created with a callback URL and a user token. This field can be returned only by the Check jobs method.

```

],
[
    "of",
    3.3,
    3.39
],
[
    "severe",
    3.39,
    3.77
],
[
    "thunderstorms",
    3.77,
    4.51
],
[
    "swept",
    4.51,
    4.79
],
[
    "through",
    4.79,
    4.95
],
[
    "Colorado",
    4.95,
    5.59

```

results

If the status is `completed`, the results of the recognition request as an array that includes a single instance of a

`SpeechRecognitionResults` object. This field is returned only by the [Check a job](#) method.

> `List[SpeechRecognitionResults]`

```
],
[
  "on",
  5.59,
  5.73
],
[
  "Sunday",
  5.73,
  6.35
],
"confidence": 0.96
}
]
}
]
}
],
"status": "completed"
}
```

warnings

List[str]

An array of warning messages about invalid parameters included with the request. Each warning includes a descriptive message and a list of invalid argument strings, for example, "unexpected query parameter 'user_token', query parameter 'callback_url' was not specified" . The request succeeds despite the warnings. This field can be returned only by the [Create a job](#) method. (If you use the `character_insertion_bias` parameter with a previous-generation model, the warning message refers to the parameter as `lambdaBias` .).

Status Code

200

OK. The request succeeded.

404	Not Found. The specified job ID was not found.
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Delete a job

Deletes the specified job. You cannot delete a job that the service is actively processing. Once you delete a job, its results are no longer available. The service automatically deletes a job and its results when the time to live for the results expires. You must use credentials for the instance of the service that owns a job to delete it.

See also: [Deleting a job](#) .

Request

parameters

```
delete_job(  
    self,  
    id: str,  
    **kwargs,  
) -> DetailedResponse
```

Example request for IBM Cloud

id

Required *
str

The identifier of the asynchronous job that is to be used for the request. You must make the request with credentials for the instance of the service that owns the job.

```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

speech_to_text.delete_job(
{id})
```

Example request for IBM Cloud Pak for Data 

Response

Status Code

204	No Content. The job was successfully deleted.
400	Bad Request. The service cannot delete a job that it is actively processing: <ul style="list-style-type: none">• Unable to delete the processing job
404	Not Found. The specified job ID was not found.
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Create a custom
language model

Creates a new custom language model for a specified base model. The custom language model can be used only with the base model for which it is created. The model is owned by the instance of the service whose credentials are used to create it.

You can create a maximum of 1024 custom language models per owning credentials. The service returns an error if you attempt to create more than 1024 models. You do not lose any models, but you cannot create any more until your model count is below the limit.

Important: Effective **31 July 2023**, all previous-generation models will be removed from the service and the documentation. Most previous-generation models were deprecated on 15 March 2022. You must migrate to the equivalent next-generation model by 31 July 2023. For more information, see [Migrating to next-generation models](#) .

See also:

- [Create a custom language model](#)
- [Language support for customization](#) .

Request

parameters

```
create_language_model(  
    self,  
    name: str,  
    base_model_name:  
    str,  
    *,  
    dialect: str =  
    None,  
    description: str =  
    None,  
    **kwargs,  
) -> DetailedResponse
```

Example request for IBM Cloud

name

Required *

str

A user-defined name for the new custom language model. Use a localized name that matches the language of the custom model. Use a name that describes the domain of the custom model, such as `Medical custom model` or `Legal custom model`. Use a name that is unique among all custom language models that you own.

Include a maximum of 256 characters in the name. Do not use backslashes, slashes, colons, equal signs, ampersands, or question marks in the name.

base_model_name

Required *

str

The name of the base language model that is to be customized by the new custom language model. The new custom model can be used only with the base model that it customizes.

To determine whether a base

```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

language_model =
speech_to_text.create_lang
uage_model(
    'First example
language model',
    'en-
US_BroadbandModel',
    description='First
custom language model
example'
).get_result()
print(json.dumps(language_
```

model supports language
model customization, use
the [Get a model](#) method and
check that the attribute
`custom_language_model`
is set to `true` . You can also
refer to [Language support for
customization](#) .

Allowable values: [

`ar-MS_Telephony` ,
`cs-CZ_Telephony` ,
`de-DE_BroadbandModel` ,
`de-DE_Multimedia` ,
`de-DE_NarrowbandModel`
, `de-DE_Telephony` ,
`en-AU_BroadbandModel` ,
`en-AU_Multimedia` ,
`en-AU_NarrowbandModel`
, `en-AU_Telephony` ,
`en-GB_BroadbandModel` ,
`en-GB_Multimedia` ,
`en-GB_NarrowbandModel`
, `en-GB_Telephony` ,
`en-IN_Telephony` ,
`en-US_BroadbandModel` ,
`en-US_Multimedia` ,
`en-US_NarrowbandModel`

```
model, indent=2))
```

Example request for IBM Cloud Pak for Data 

```
,
en-
US_ShortForm_NarrowbandModel
, en-US_Telephony ,
en-
WW_Medical_Telephony
, es-AR_BroadbandModel ,
es-AR_NarrowbandModel
, es-CL_BroadbandModel ,
es-CL_NarrowbandModel
, es-CO_BroadbandModel ,
es-CO_NarrowbandModel
, es-ES_BroadbandModel ,
es-ES_NarrowbandModel
, es-ES_Multimedia ,
es-ES_Telephony ,
es-LA_Telephony ,
es-MX_BroadbandModel ,
es-MX_NarrowbandModel
, es-PE_BroadbandModel ,
es-PE_NarrowbandModel
, fr-CA_BroadbandModel ,
fr-CA_Multimedia ,
fr-CA_NarrowbandModel
, fr-CA_Telephony ,
fr-FR_BroadbandModel ,
fr-FR_Multimedia ,
```

```
fr-FR_NarrowbandModel
, fr-FR_Telephony ,
hi-IN_Telephony ,
it-IT_BroadbandModel ,
it-IT_NarrowbandModel
, it-IT_Multimedia ,
it-IT_Telephony ,
ja-JP_BroadbandModel ,
ja-JP_Multimedia ,
ja-JP_NarrowbandModel
, ja-JP_Telephony ,
ko-KR_BroadbandModel ,
ko-KR_Multimedia ,
ko-KR_NarrowbandModel
, ko-KR_Telephony ,
nl-BE_Telephony ,
nl-NL_BroadbandModel ,
nl-NL_Multimedia ,
nl-NL_NarrowbandModel
, nl-NL_Telephony ,
pt-BR_BroadbandModel ,
pt-BR_Multimedia ,
pt-BR_NarrowbandModel
, pt-BR_Telephony ,
sv-SE_Telephony ,
zh-CN_Telephony ]
```

dialect

str

The dialect of the specified language that is to be used with the custom language model. *For all languages, it is always safe to omit this field.*

The service automatically uses the language identifier from the name of the base model. For example, the service automatically uses **en-US** for all US English models.

If you specify the **dialect** for a new custom model, follow these guidelines. *For non-Spanish previous-generation models and for next-generation models*, you must specify a value that matches the five-character language identifier from the name of the base model. *For Spanish previous-generation models*, you must specify one of the following values:

- **es-ES** for Castilian Spanish (**es-ES** models)
- **es-LA** for Latin American Spanish (**es-AR** , **es-CL** , **es-CO** ,

and `es-PE` models)

- `es-US` for Mexican (North American) Spanish (`es-MX` models)

All values that you pass for the `dialect` field are case-insensitive.

description

str

A recommended description of the new custom language model. Use a localized description that matches the language of the custom model. Include a maximum of 128 characters in the description.

Response

LanguageModel

Information about an existing custom language model.

Example responses

Success example



customization_id

Always included *

str

The customization ID (GUID) of the custom language model. The [Create a custom language model](#) method returns only this field of the object; it does not return the other fields.

created

str

The date and time in Coordinated Universal Time (UTC) at which the custom language model was created. The value is provided in full ISO 8601 format (`YYYY-MM-DDThh:mm:ss.sTZD`).

```
{  
  "customization_id":  
    "74f4807e-b5ff-4866-824e-  
    6bba1a84fe96"  
}
```

updated

str

The date and time in Coordinated Universal Time (UTC) at which the custom language model was last modified. The `created` and `updated` fields are equal when a language model is first added but has yet to be updated. The value is provided in full ISO 8601 format (YYYY-MM-DDThh:mm:ss.STZD).

language

str

The language identifier of the custom language model (for example, `en-US`). The value matches the five-character language identifier from the name of the base model for the custom model. This value might be different from the value of the `dialect` field.

dialect

str

The dialect of the language for the custom language model. *For custom models that are based on non-Spanish previous-generation models and on next-generation models*, the field

matches the language of the base model; for example, `en-US` for one of the US English models. *For custom models that are based on Spanish previous-generation models*, the field indicates the dialect with which the model was created. The value can match the name of the base model or, if it was specified by the user, can be one of the following:

- `es-ES` for Castilian Spanish (`es-ES` models)
- `es-LA` for Latin American Spanish (`es-AR` , `es-CL` , `es-CO` , and `es-PE` models)
- `es-US` for Mexican (North American) Spanish (`es-MX` models)

Dialect values are case-insensitive.

versions List[str]	A list of the available versions of the custom language model. Each element of the array indicates a version of the base model with which the custom model can be used. Multiple versions exist only if the custom model has been upgraded to a new version of its base model. Otherwise, only a single version is shown.
------------------------------	---

owner str	The GUID of the credentials for the instance of the service that owns the custom language model.
---------------------	--

name str	The name of the custom language model.
--------------------	--

description str	The description of the custom language model.
---------------------------	---

base_model_name str	The name of the language model for which the custom language model was created.
-------------------------------	---

status

str

The current status of the custom language model:

- **pending** : The model was created but is waiting either for valid training data to be added or for the service to finish analyzing added data.
- **ready** : The model contains valid data and is ready to be trained. If the model contains a mix of valid and invalid resources, you need to set the **strict** parameter to **false** for the training to proceed.
- **training** : The model is currently being trained.
- **available** : The model is trained and ready to use.
- **upgrading** : The model is currently being upgraded.
- **failed** : Training of the model failed.

Possible values: [**pending** ,

`ready , training ,
available , upgrading ,
failed]`

progress

int

A percentage that indicates the progress of the custom language model's current training. A value of `100` means that the model is fully trained. **Note:** The `progress` field does not currently reflect the progress of the training. The field changes from `0` to `100` when training is complete.

error

str

If an error occurred while adding a grammar file to the custom language model, a message that describes an `Internal Server Error` and includes the string `Cannot compile grammar` . The status of the custom model is not affected by the error, but the grammar cannot be used with the model.

warnings

str

If the request included unknown parameters, the following message:

```
Unexpected query
parameter(s)
['parameters']
detected , where
parameters is a list that
includes a quoted string for
each unknown parameter.
```

Status Code

201

Created. The custom language model was successfully created.

400

Bad Request. A required parameter is null or invalid. Specific failure messages include:

- Required parameter '{name}' is missing
- Required parameter '{name}' cannot be empty string
- Required parameter '{name}' cannot be null
- The base model '{model_name}' is not recognized

- Language customization is not supported for base model '{model_name}'
- Invalid dialect value '{dialect}' specified for language '{language}'
- You exceeded the maximum '{model_number}' of allowed custom language models. You have '{model_number}' custom language models. Please remove the models you do not need or contact the IBM speech support team to apply for an exception.

401 **Unauthorized.** The specified credentials are invalid.

500 **Internal Server Error.** The service experienced an internal error.

503 **Service Unavailable.** The service is currently unavailable.

List custom language

models

Lists information about all custom language models that are owned by an instance of the service. Use the `language` parameter to see all custom language models for the specified language. Omit the parameter to see all custom language models for all languages. You must use credentials for the instance of the service that owns a model to list information about it.

See also:

- [Listing custom language models](#)
- [Language support for customization](#) .

Request

parameters	
language str	The identifier of the language for which custom language or custom acoustic models are to be returned. Specify the five-character language identifier; for example, specify <code>en-US</code> to see all custom language or custom acoustic models that are

```
list_language_models(  
    self,  
    *,  
    language: str =  
None,  
    **kwargs,  
) -> DetailedResponse
```

Example request for IBM Cloud

based on US English models.
Omit the parameter to see all custom language or custom acoustic models that are owned by the requesting credentials.

To determine the languages for which customization is available, see [Language support for customization](#).

Allowable values: [`ar-MS` ,
`cs-CZ` , `de-DE` , `en-AU` ,
`en-GB` , `en-IN` , `en-US` ,
`en-WW` , `es-AR` , `es-CL` ,
`es-CO` , `es-ES` , `es-LA` ,
`es-MX` , `es-PE` , `fr-CA` ,
`fr-FR` , `hi-IN` , `it-IT` ,
`ja-JP` , `ko-KR` , `nl-BE` ,
`nl-NL` , `pt-BR` , `sv-SE` ,
`zh-CN`]

```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

language_models =
speech_to_text.list_langua
ge_models().get_result()
print(json.dumps(language_
models, indent=2))
```

Example request for IBM Cloud Pak for Data

Response

LanguageModels Information about existing custom language models.

Example responses

customizations An array of
Always included * `LanguageModel` objects that provides information about each available custom language model. The array is empty if the requesting credentials own no custom language models (if no language is specified) or own no custom language models for the specified language.

> [List\[LanguageModel\]](#)

Status Code

200 OK. The request succeeded.

400 **Bad Request.** A required parameter is null or invalid. Specific failure messages include:

- `Language '{language}' is not supported for customization.`

Success example

```
{
  "customizations": [
    {
      "customization_id":
"74f4807e-b5ff-4866-824e-6bba1a84fe96",
      "created": "2016-06-01T14:21:26.894Z",
      "updated": "2020-01-18T18:42:25.324Z",
      "language": "en-US",
      "dialect": "en-US",
      "versions": [
        "en-US_BroadbandModel.v2018-07-31",
        "en-US_BroadbandModel.v2020-01-16"
      ],
      "owner": "297cfd08-330a-22ba-93ce-1a73f454dd98",
      "name": "Example model",
      "description": "Example custom language"
```

401	Unauthorized. The specified credentials are invalid.
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

```
model",
  "base_model_name":
"en-US_BroadbandModel",
  "status": "pending",
  "progress": 0
},
{
  "customization_id":
"8391f918-3b76-e109-763c-
b7732fae4829",
  "created": "2017-12-
02T18:51:37.291Z",
  "updated": "2017-12-
02T20:02:10.624Z",
  "language": "en-US",
  "dialect": "en-US",
  "versions": [
    "en-
US_BroadbandModel.v2017-
11-15"
  ],
  "owner": "297cfd08-
330a-22ba-93ce-
1a73f454dd98",
  "name": "Example
model two",
  "description":
"Example custom language
model two",
  "base_model_name":
"en-US_BroadbandModel",
  "status":
"available",
  "progress": 100
```

```
}  
]  
}
```

Get a custom language model

Gets information about a specified custom language model. You must use credentials for the instance of the service that owns a model to list information about it.

See also:

- [Listing custom language models](#)
- [Language support for customization](#) .

Request

parameters

```
get_language_model(  
    self,  
    customization_id:  
str,  
    **kwargs,  
) -> DetailedResponse
```



Example request for IBM Cloud




customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.



```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

language_model =
speech_to_text.get_languag
e_model('{customization_id
}').get_result()
print(json.dumps(language_
model, indent=2))
```

Example request for IBM Cloud Pak for Data 

Response

LanguageModel Information about an existing custom language model.

Example responses

Success example

customization_id

Always included *

str

The customization ID (GUID) of the custom language model. The [Create a custom language model](#) method returns only this field of the object; it does not return the other fields.

created

str

The date and time in Coordinated Universal Time (UTC) at which the custom language model was created. The value is provided in full ISO 8601 format (**YYYY-MM-DDThh:mm:ss.sTZD**).

updated

str

The date and time in Coordinated Universal Time (UTC) at which the custom language model was last modified. The `created` and `updated` fields are equal when a language model is first added but has yet to be updated. The value is provided in full ISO 8601 format (YYYY-MM-DDThh:mm:ss.STZD).

language

str

The language identifier of the custom language model (for example, `en-US`). The value matches the five-character language identifier from the name of the base model for the custom model. This value might be different from the value of the `dialect` field.

dialect

str

The dialect of the language for the custom language model. *For custom models that are based on non-Spanish previous-generation models and on next-generation models*, the field

```
{
  "customization_id":
  "74f4807e-b5ff-4866-824e-6bba1a84fe96",
  "created": "2016-06-01T14:21:26.894Z",
  "updated": "2020-01-18T18:42:25.324Z",
  "language": "en-US",
  "dialect": "en-US",
  "versions": [
    "en-US_BroadbandModel.v2018-07-31",
    "en-US_BroadbandModel.v2020-01-16"
  ],
  "owner": "297cfd08-330a-22ba-93ce-1a73f454dd98",
  "name": "Example model",
  "description": "Example custom language model",
  "base_model_name": "en-US_BroadbandModel",
  "status": "pending",
  "progress": 0
}
```



matches the language of the base model; for example, `en-US` for one of the US English models. *For custom models that are based on Spanish previous-generation models*, the field indicates the dialect with which the model was created. The value can match the name of the base model or, if it was specified by the user, can be one of the following:

- `es-ES` for Castilian Spanish (`es-ES` models)
- `es-LA` for Latin American Spanish (`es-AR` , `es-CL` , `es-CO` , and `es-PE` models)
- `es-US` for Mexican (North American) Spanish (`es-MX` models)

Dialect values are case-insensitive.

versions List[str]	A list of the available versions of the custom language model. Each element of the array indicates a version of the base model with which the custom model can be used. Multiple versions exist only if the custom model has been upgraded to a new version of its base model. Otherwise, only a single version is shown.
------------------------------	---

owner str	The GUID of the credentials for the instance of the service that owns the custom language model.
---------------------	--

name str	The name of the custom language model.
--------------------	--

description str	The description of the custom language model.
---------------------------	---

base_model_name str	The name of the language model for which the custom language model was created.
-------------------------------	---

status

str

The current status of the custom language model:

- **pending** : The model was created but is waiting either for valid training data to be added or for the service to finish analyzing added data.
- **ready** : The model contains valid data and is ready to be trained. If the model contains a mix of valid and invalid resources, you need to set the **strict** parameter to **false** for the training to proceed.
- **training** : The model is currently being trained.
- **available** : The model is trained and ready to use.
- **upgrading** : The model is currently being upgraded.
- **failed** : Training of the model failed.

Possible values: [**pending** ,

`ready , training ,
available , upgrading ,
failed]`

progress

int

A percentage that indicates the progress of the custom language model's current training. A value of `100` means that the model is fully trained. **Note:** The `progress` field does not currently reflect the progress of the training. The field changes from `0` to `100` when training is complete.

error

str

If an error occurred while adding a grammar file to the custom language model, a message that describes an `Internal Server Error` and includes the string `Cannot compile grammar` . The status of the custom model is not affected by the error, but the grammar cannot be used with the model.

warnings

str

If the request included unknown parameters, the following message:

```
Unexpected query
parameter(s)
['parameters']
detected , where
parameters is a list that
includes a quoted string for
each unknown parameter.
```

Status Code

200

OK. The request succeeded.

400

Bad Request. The specified customization ID is invalid:

- `Malformed GUID:`
`'{customization_id}'`

401

Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials:

- `Invalid customization_id`
`'{customization_id}'` for
`user`

500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Delete a custom language model

Deletes an existing custom language model. The custom model cannot be deleted if another request, such as adding a corpus or grammar to the model, is currently being processed. You must use credentials for the instance of the service that owns a model to delete it.

See also:

- [Deleting a custom language model](#)
- [Language support for customization](#) .

Request

parameters

```
delete_language_model(
    self,
    customization_id:
str,
    **kwargs,
) -> DetailedResponse
```

Example request for IBM Cloud



customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.



```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

speech_to_text.delete_lang
uage_model('{customization
_id}')
```

Example request for IBM Cloud Pak for Data 

Response

Status Code

Example responses

200	OK. The custom language model was successfully deleted.
400	Bad Request. The specified customization ID is invalid: <ul style="list-style-type: none">Malformed GUID: <code>'{customization_id}'</code>
401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials, including the case where the custom model does not exist: <ul style="list-style-type: none">Invalid customization_id <code>'{customization_id}'</code> for user
409	Conflict. The service is currently busy handling a previous request for the custom model: <ul style="list-style-type: none">Customization <code>'{customization_id}'</code> is currently locked to process your last request.
500	Internal Server Error. The service experienced an internal error.

Success example

```
{}
```

Train a custom language model

Initiates the training of a custom language model with new resources such as corpora, grammars, and custom words. After adding, modifying, or deleting resources for a custom language model, use this method to begin the actual training of the model on the latest data. You can specify whether the custom language model is to be trained with all words from its words resource or only with words that were added or modified by the user directly. You must use credentials for the instance of the service that owns a model to train it.

The training method is asynchronous. It can take on the order of minutes to complete depending on the amount of data on which the service is being trained and the current load on the service. The method returns an HTTP 200 response code to indicate that the training process has begun.

You can monitor the status of the training by using the [Get a custom language model](#)

```
train_language_model(  
    self,  
    customization_id:  
    str,  
    *,  
    word_type_to_add:  
    str = None,  
  
    customization_weight:  
    float = None,  
    strict: bool =  
    None,  
    force: bool =  
    None,  
    **kwargs,  
) -> DetailedResponse
```

method to poll the model's status. Use a loop to check the status every 10 seconds. If you added custom words directly to a custom model that is based on a next-generation model, allow for some minutes of extra training time for the model.

The method returns a `LanguageModel` object that includes `status` and `progress` fields. A status of `available` means that the custom model is trained and ready to use. The service cannot accept subsequent training requests or requests to add new resources until the existing request completes.

For custom models that are based on improved base language models, training also performs an automatic upgrade to a newer version of the base model. You do not need to use the [Upgrade a custom language model](#) method to perform the upgrade.

See also:

- [Language support for customization](#)
- [Train the custom language model](#)
- [Upgrading custom language models that are based on improved next-generation models](#)

Training failures

- Training can fail to start for the following reasons:
- The service is currently handling another request for the custom model, such as another training request or a request to add a corpus or grammar to the model.
 - No training data have been added to the custom model.
 - The custom model contains one or more invalid corpora, grammars, or words (for example, a custom word has an invalid sounds-like pronunciation). You can correct the invalid resources or set the `strict` parameter to `false` to exclude the invalid resources from the training. The model must contain at least one valid resource for training to succeed.

Request

parameters

Example request for IBM Cloud



customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

word_type_to_add

str

For custom models that are based on previous-generation models, the type of words from the custom language model's words resource on which to train the model:

- **all** (the default) trains the model on all new words, regardless of whether they were extracted from corpora or grammars or were added or modified by the user.
- **user** trains the model

```
from ibm_watson import  
SpeechToTextV1  
from  
ibm_cloud_sdk_core.authent  
icators import  
IAMAuthenticator
```

```
authenticator =  
IAMAuthenticator('{apikey}  
' )  
speech_to_text =  
SpeechToTextV1(
```

```
authenticator=authenticato  
r  
)
```

```
speech_to_text.set_service  
_url('{url}')
```

```
speech_to_text.train_langu  
age_model('{customization_  
id}')  
# Poll for language model  
status.
```

Example request for IBM Cloud Pak for Data 

only on custom words that were added or modified by the user directly. The model is not trained on new words extracted from corpora or grammars.

For custom models that are based on next-generation models, the service ignores the parameter. The words resource contains only custom words that the user adds or modifies directly, so the parameter is unnecessary.

Allowable values: [[all](#) , [user](#)]

Default: [all](#)

customization_weight

float

Specifies a customization weight for the custom language model. The customization weight tells the service how much weight to give to words from the custom language model compared to those from the

base model for speech recognition. Specify a value between 0.0 and 1.0. The default value is:

- 0.3 for previous-generation models
- 0.2 for most next-generation models
- 0.1 for next-generation English and Japanese models

The default value yields the best performance in general. Assign a higher value if your audio makes frequent use of OOV words from the custom model. Use caution when setting the weight: a higher value can improve the accuracy of phrases from the custom model's domain, but it can negatively affect performance on non-domain phrases.

The value that you assign is used for all recognition requests that use the model. You can override it for any recognition request by specifying a customization

weight for that request.

See [Using customization weight](#) .

strict

bool

If `false` , allows training of the custom language model to proceed as long as the model contains at least one valid resource. The method returns an array of `TrainingWarning` objects that lists any invalid resources. By default (`true`), training of a custom language model fails (status code 400) if the model contains one or more invalid resources (corpus files, grammar files, or custom words).

Default: `true`

force

bool

If `true` , forces the training of the custom language model regardless of whether it contains any changes (is in the `ready` or `available` state). By default (`false`), the model must be in the `ready` state to be trained. You can use the parameter to train and thus upgrade a custom model that is based on an improved next-generation model. *The parameter is available only for IBM Cloud, not for IBM Cloud Pak for Data.*

See [Upgrading a custom language model based on an improved next-generation model](#) .

Default: `false`

Response

TrainingResponseThe response from training of a custom language or custom acoustic model.

Example responses

Success example



warnings

An array of `TrainingWarning` objects that lists any invalid resources contained in the custom model. For custom language models, invalid resources are grouped and identified by type of resource. The method can return warnings only if the `strict` parameter is set to `false`.

> [List\[TrainingWarning\]](#)

`{}`



Status Code

200	OK. Training of the custom language model started successfully.
------------	--

400

Bad Request. A required parameter is null or invalid, the custom model is not ready to be trained, or the total number of words or OOV words exceeds the maximum threshold. Specific failure messages include:

- `No input data available for running training`
- `Fix errors in the following words: [{words}] before training`
- `Total number of words {number} exceeds maximum allowed`
- `Total number of OOV words {number} exceeds {maximum}`
- `Malformed GUID: '{customization_id}'`

401

Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials:

- `Invalid customization_id '{customization_id}' for user`
-

409

Conflict. The service is currently busy handling a previous request for the custom model:

- Customization
'{customization_id}' is currently locked to process your last request.

500

Internal Server Error. The service experienced an internal error.

503

Service Unavailable. The service is currently unavailable.

Reset a custom
language model

Resets a custom language model by removing all corpora, grammars, and words from the model. Resetting a custom language model initializes the model to its state when it was first created. Metadata such as the name and language of the model are preserved, but the model's words resource is removed and must be re-created. You must use credentials for the instance of the service that owns a model to reset it.

See also:

- [Resetting a custom language model](#)
- [Language support for customization](#) .

Request

parameters

```
reset_language_model(  
    self,  
    customization_id:  
str,  
    **kwargs,  
) -> DetailedResponse
```



Example request for IBM Cloud



customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

```
from ibm_watson import  
SpeechToTextV1  
from  
ibm_cloud_sdk_core.authent  
icators import  
IAMAuthenticator
```

```
authenticator =  
IAMAuthenticator('{apikey}'  
)  
speech_to_text =  
SpeechToTextV1(  

```

```
authenticator=authenticato  
r  
)
```

```
speech_to_text.set_service  
_url('{url}')
```

```
speech_to_text.reset_langu  
age_model('{customization_  
id}')
```

Example request for IBM Cloud Pak for Data

Response

Status Code

Example responses

200	OK. The custom language model was successfully reset.
400	Bad Request. The specified customization ID is invalid: <ul style="list-style-type: none">Malformed GUID: '{customization_id}'
401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none">Invalid customization_id '{customization_id}' for user
409	Conflict. The service is currently busy handling a previous request for the custom model: <ul style="list-style-type: none">Customization '{customization_id}' is currently locked to process your last request.
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Success example

```
{}
```

Upgrade a custom language model

Initiates the upgrade of a custom language model to the latest version of its base language model. The upgrade method is asynchronous. It can take on the order of minutes to complete depending on the amount of data in the custom model and the current load on the service. A custom model must be in the `ready` or `available` state to be upgraded. You must use credentials for the instance of the service that owns a model to upgrade it.

The method returns an HTTP 200 response code to indicate that the upgrade process has begun successfully. You can monitor the status of the upgrade by using the [Get a custom language model](#) method to poll the model's status. The method returns a `LanguageModel` object that includes `status` and `progress` fields. Use a loop to check the status every 10 seconds.

While it is being upgraded, the custom model has the status `upgrading`. When the upgrade is complete, the model resumes the status that it had prior to upgrade. The service cannot

```
upgrade_language_model(  
    self,  
    customization_id:  
str,  
    **kwargs,  
) -> DetailedResponse
```



accept subsequent requests for the model until the upgrade completes.

For custom models that are based on improved base language models, the [Train a custom language model](#) method also performs an automatic upgrade to a newer version of the base model. You do not need to use the upgrade method.

See also:

- [Language support for customization](#)
- [Upgrading a custom language model](#)
- [Upgrading custom language models that are based on improved next-generation models](#)

Request

parameters

Example request for IBM Cloud



customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator
```

```
authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(
```

```
authenticator=authenticato
r
)
```

```
speech_to_text.set_service
_url('{url}')
```

```
speech_to_text.upgrade_lan
guage_model('{customizatio
n_id}')
# Poll for language model
status.
```

Example request for IBM Cloud Pak for Data 

Response

Status Code

Status code		Example responses
200	OK. Upgrade of the custom language model started successfully.	<div>Success example</div> <div><pre>{}</pre></div>
400	Bad Request. The specified customization ID is invalid or the specified custom model cannot be upgraded: <ul style="list-style-type: none">Malformed GUID: <code>'{customization_id}'</code>Custom model is up-to-dateNo input data available to upgrade the modelCannot upgrade failed custom model	
401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none">Invalid customization_id <code>'{customization_id}'</code> for user	

409	Conflict. The service is currently busy handling a previous request for the custom model: <ul style="list-style-type: none">Customization '{customization_id}' is currently locked to process your last request.
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

List corpora

Lists information about all corpora from a custom language model. The information includes the name, status, and total number of words for each corpus. *For custom models that are based on previous-generation models*, it also includes the number of out-of-vocabulary (OOV) words from the corpus. You must use credentials for the instance of the service that owns a model to list its corpora.

See also: [Listing corpora for a custom language model](#) .

Request


parameters
<div><div>customization_id</div><div>Required *</div><div>str</div><div>The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.</div></div>

```
list_corpora(  
    self,  
    customization_id:  
str,  
    **kwargs,  
) -> DetailedResponse
```



Example request for IBM Cloud





```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)


speech_to_text.set_service
_url('{url}')

corpora =
speech_to_text.list_corpor
a('{customization_id'}).ge
t_result()
print(json.dumps(corpora,
indent=2))
```

Example request for IBM Cloud Pak for Data 

Response

Corpora	Information about the corpora from a custom language model.
corpora Always included *	An array of Corpus objects that provides information about the corpora for the custom model. The array is empty if the custom model has no corpora. > List[Corpus]
Status Code	
200	OK. The request succeeded.
400	Bad Request. The specified customization ID is invalid: <ul style="list-style-type: none">Malformed GUID: <code>'{customization_id}'</code>

Example responses	
Success example	
<pre>{ "corpora": [{ "name": "corpus1", "out_of_vocabulary_words": 191, "total_words": 5037, "status": "analyzed" }, { "name": "corpus2", "out_of_vocabulary_words": 0, "total_words": 0, "status": "being_processed" }, { "name": "corpus3", "out_of_vocabulary_words": 0, "total_words": 0, "status":</pre>	

401 Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials:

- `Invalid customization_id '{customization_id}' for user`

500 Internal Server Error. The service experienced an internal error.

503 Service Unavailable. The service is currently unavailable.

```
"undetermined",
    "error": "Analysis
of corpus 'corpus3.txt'
failed. Please try adding
the corpus again by
setting the
'allow_overwrite' flag to
'true'."
}
]
}
```

Add a corpus

Adds a single corpus text file of new training data to a custom language model. Use multiple requests to submit multiple corpus text files. You must use credentials for the instance of the service that owns a model to add a corpus to it. Adding a corpus does not affect the custom language model until you train the model for the new data by using the [Train a custom language model](#) method.

Submit a plain text file that contains sample sentences from the domain of interest to enable the service to parse the words in

```
add_corpus(
    self,
    customization_id:
str,
    corpus_name: str,
    corpus_file:
BinaryIO,
    *,
    allow_overwrite:
bool = None,
    **kwargs,
) -> DetailedResponse
```

context. The more sentences you add that represent the context in which speakers use words from the domain, the better the service's recognition accuracy.

The call returns an HTTP 201 response code if the corpus is valid. The service then asynchronously processes and automatically extracts data from the contents of the corpus. This operation can take on the order of minutes to complete depending on the current load on the service, the total number of words in the corpus, and, *for custom models that are based on previous-generation models*, the number of new (out-of-vocabulary) words in the corpus. You cannot submit requests to add additional resources to the custom model or to train the model until the service's analysis of the corpus for the current request completes. Use the [Get a corpus](#) method to check the status of the analysis.

For custom models that are based on previous-generation models, the service auto-populates the model's words resource with words from the corpus that are not found in its base vocabulary. These words are referred to as out-of-vocabulary (OOV) words. After adding a corpus, you must validate the words resource to ensure that each OOV word's definition is complete and valid. You can use the [List custom words](#) method to examine the words resource. You can use other words method to

eliminate typos and modify how words are pronounced and displayed as needed.

To add a corpus file that has the same name as an existing corpus, set the `allow_overwrite` parameter to `true` ; otherwise, the request fails. Overwriting an existing corpus causes the service to process the corpus text file and extract its data anew. *For a custom model that is based on a previous-generation model*, the service first removes any OOV words that are associated with the existing corpus from the model's words resource unless they were also added by another corpus or grammar, or they have been modified in some way with the [Add custom words](#) or [Add a custom word](#) method.

The service limits the overall amount of data that you can add to a custom model to a maximum of 10 million total words from all sources combined. *For a custom model that is based on a previous-generation model*, you can add no more than 90 thousand custom (OOV) words to a model. This includes words that the service extracts from corpora and grammars, and words that you add directly.

See also:

- [Add a corpus to the custom language model](#)
- [Working with corpora for previous-generation models](#)

- [Working with corpora for next-generation models](#)
- [Validating a words resource for previous-generation models](#)
- [Validating a words resource for next-generation models](#)

Request

parameters

customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

corpus_name

Required *

str

The name of the new corpus for the custom language model. Use a localized name that matches the language of the custom model and reflects the contents of the

Example request for IBM Cloud

```
from os.path import join,
dirname
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)
```


corpus.

- Include a maximum of 128 characters in the name.
- Do not use characters that need to be URL-encoded. For example, do not use spaces, slashes, backslashes, colons, ampersands, double quotes, plus signs, equals signs, questions marks, and so on in the name. (The service does not prevent the use of these characters. But because they must be URL-encoded wherever used, their use is strongly discouraged.)
- Do not use the name of an existing corpus or grammar that is already defined for the custom model.
- Do not use the name `user`, which is reserved by the service to denote custom words that are added or modified by the

```
speech_to_text.set_service_url('{url}')
```

```
with  
open(join(dirname(__file__  
) , './.', 'corpus1.txt'),  
      'rb') as
```

```
corpus_file:
```

```
speech_to_text.add_corpus(  
  
    '{customization_id}',  
    'corpus1',  
    corpus_file  
)  
# Poll for corpus status.
```

Example request for IBM Cloud Pak for Data 

user.

- Do not use the name `base_lm` or `default_lm`. Both names are reserved for future use by the service.

corpus_file

Required *

BinaryIO

A plain text file that contains the training data for the corpus. Encode the file in UTF-8 if it contains non-ASCII characters; the service assumes UTF-8 encoding if it encounters non-ASCII characters.

Make sure that you know the character encoding of the file. You must use that same encoding when working with the words in the custom language model. For more information, see [Character encoding for custom words](#).

With the `curl` command, use the `--data-binary` option to upload the file for the request.

allow_overwrite If `true` , the specified bool corpus overwrites an existing corpus with the same name. If `false` , the request fails if a corpus with the same name already exists. The parameter has no effect if a corpus with the same name does not already exist.

Default: `false`

Response

Status Code

- | | |
|-----|--|
| 201 | Created. Addition of the corpus data was successfully started. The service is analyzing the data. |
| 400 | Bad Request. A required parameter is null or invalid, the specified corpus name already exists, or the custom model needs to be upgraded, among other possibilities. Specific failure messages include: <ul style="list-style-type: none">Malformed GUID: <code>'{customization_id}'</code>Corpus file not |

Example responses

Success example

`{}`

specified or empty

- Corpus '{corpus_name}' already exists - change its name, remove existing file before adding new one, or overwrite existing file by setting 'allow_overwrite' to 'true'
- Grammar exists with corpus name '{corpus_name}'. Please use different name.
- TOTAL_NUMBER_OF_OOV_WORDS_EXCEEDS_MAXIMUM_ALLOWED_FORMAT: "Total number of OOV words {total_number} exceeds {maximum_allowed}"
- Analysis of corpus '{corpus_name}' failed due to {error_message}. Please fix the error then add the corpus again by setting the 'allow_overwrite' flag to 'true'. , where {error_message} is a message of the form {"code": 404, "error": "Model en-

```
US_BroadbandModel
(version: en-
US_BroadbandModel.
{version}) not found",
"code_description": "Not
Found"} . Upgrade the custom
language model to the latest
version of its base language
model, and then add the corpus
to the custom model.
```

401 Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials:

- `Invalid customization_id '{customization_id}' for user`

405 Method Not Allowed. The corpus name includes characters that need to be URL-encoded.

409	<p>Conflict. The service is currently busy handling a previous request for the custom model:</p> <ul style="list-style-type: none">• Customization '{customization_id}' is currently locked to process your last request.
415	<p>Unsupported Media Type. The request specified an unacceptable media type.</p>
500	<p>Internal Server Error. An internal error prevented the service from satisfying the request. You can also receive status code 500 Forwarding Error if the service is currently busy handling a previous request for the custom model.</p>
503	<p>Service Unavailable. The service is currently unavailable.</p>

Get a corpus

Gets information about a corpus from a custom language model. The information includes the name, status, and total number of words for the corpus. *For custom models that are based on previous-generation models*, it also includes the number of out-of-vocabulary (OOV) words from the corpus. You must use credentials for the instance of the service that owns a model to list its corpora.

See also: [Listing corpora for a custom language model](#) .

Request

parameters

customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

```
get_corpus(  
    self,  
    customization_id:  
    str,  
    corpus_name: str,  
    **kwargs,  
) -> DetailedResponse
```



Example request for IBM Cloud



corpus_name

Required *

str

The name of the corpus for
the custom language model.

```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

corpus =
speech_to_text.get_corpus(
    '{customization_id}',
    'corpus1'
).get_result()
print(json.dumps(corpus,
indent=2))
```

Example request for IBM Cloud Pak for Data 

Response

Corpus	Information about a corpus from a custom language model.
name Always included * str	The name of the corpus.
total_words Always included * int	The total number of words in the corpus. The value is 0 while the corpus is being processed.

Example responses

Success example

```
{
  "name": "corpus1",
  "out_of_vocabulary_words": 191,
  "total_words": 5037,
  "status": "analyzed"
}
```

out_of_vocabulary_words

Always included *

int

For custom models that are based on previous-generation models, the number of OOV words extracted from the corpus.

The value is 0 while the corpus is being processed.

For custom models that are based on next-generation models, no OOV words are extracted from corpora, so the value is always 0 .

status

Always included *
str

The status of the corpus:

- **analyzed** : The service successfully analyzed the corpus. The custom model can be trained with data from the corpus.
- **being_processed** : The service is still analyzing the corpus. The service cannot accept requests to add new resources or to train the custom model.
- **undetermined** : The service encountered an error while processing the corpus. The **error** field describes the failure.

Possible values: [

analyzed ,
being_processed ,
undetermined]

error

str

If the status of the corpus is

`undetermined` , the

following message:

```
Analysis of corpus
'name' failed. Please
try adding the corpus
again by setting the
'allow_overwrite' flag
to 'true'.
```

Status Code**200****OK.** The request succeeded.**400**

Bad Request. The specified customization ID or corpus name is invalid, including the case where the corpus does not exist for the custom model. Specific failure messages include:

- `Malformed GUID:`
`'{customization_id}'`
- `Invalid value for corpus`
`name '{corpus_name}'`

401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none">• <code>Invalid customization_id '{customization_id}' for user</code>
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Delete a corpus

Deletes an existing corpus from a custom language model. Removing a corpus does not affect the custom model until you train the model with the [Train a custom language model](#) method. You must use credentials for the instance of the service that owns a model to delete its corpora.

For custom models that are based on previous-generation models, the service removes any out-of-vocabulary (OOV) words that are associated with the corpus from the custom model's words resource unless they were also added by another corpus or grammar, or they were modified in some way with the [Add custom words](#) or [Add a custom word](#) method.

See also: [Deleting a corpus from a custom language model](#) .

Request

parameters

```
delete_corpus(  
    self,  
    customization_id:  
    str,  
    corpus_name: str,  
    **kwargs,  
) -> DetailedResponse
```

Example request for IBM Cloud

customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

corpus_name

Required *

str

The name of the corpus for the custom language model.

```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator
```

```
authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(
```

```
authenticator=authenticato
r
)
```

```
speech_to_text.set_service
_url('{url}')
```

```
speech_to_text.delete_corp
us(
    '{customization_id}',
    'corpus1'
)
```

Example request for IBM Cloud Pak for Data

Response

Status Code

Status code		Example responses
200	OK. The corpus was successfully deleted from the custom language model.	<div>Success example</div> <div><pre>{}</pre></div>
400	Bad Request. The specified customization ID or corpus name is invalid, including the case where the corpus does not exist for the custom model. Specific failure messages include: <ul style="list-style-type: none">Malformed GUID: <code>'{customization_id}'</code>Invalid value for corpus name <code>'{corpus_name}'</code>	
401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none">Invalid customization_id <code>'{customization_id}'</code> for user	
405	Method Not Allowed. No corpus name was specified with the request.	

409

Conflict. The service is currently busy handling a previous request for the custom model:

- Customization
'{customization_id}' is currently locked to process your last request.

500

Internal Server Error. The service experienced an internal error.

503

Service Unavailable. The service is currently unavailable.

List custom words

Lists information about custom words from a custom language model. You can list all words from the custom model's words resource, only custom words that were added or modified by the user, or, *for a custom model that is based on a previous-generation model*, only out-of-vocabulary (OOV) words that were extracted from corpora or are recognized by grammars. *For a custom model that is based on a next-generation model*, you can list all words or only those words that were added directly by a user, which return the same results.

You can also indicate the order in which the service is to return words; by default, the service lists words in ascending alphabetical order. You must use credentials for the instance of the service that owns a model to list information about its words.

See also: [Listing words from a custom language model](#) .

Request

parameters

```
list_words(  
    self,  
    customization_id:  
    str,  
    *,  
    word_type: str =  
    None,  
    sort: str = None,  
    **kwargs,  
) -> DetailedResponse
```



Example request for IBM Cloud



customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

word_type

str

The type of words to be listed from the custom language model's words resource:

- `all` (the default) shows all words.
- `user` shows only custom words that were added or modified by the user directly.
- `corpora` shows only OOV that were extracted from corpora.
- `grammars` shows only OOV words that are recognized by grammars.

```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

words =
speech_to_text.list_words(
'{customization_id}').get_
result()
print(json.dumps(words,
indent=2))
```

Example request for IBM Cloud Pak for Data 

For a custom model that is based on a next-generation model, only `all` and `user` apply. Both options return the same results. Words from other sources are not added to custom models that are based on next-generation models.

Allowable values: [`all` ,
`user` , `corpora` ,
`grammars`]

Default: `all`

sort

str

Indicates the order in which the words are to be listed, [alphabetical](#) or by [count](#) . You can prepend an optional [+](#) or [-](#) to an argument to indicate whether the results are to be sorted in ascending or descending order. By default, words are sorted in ascending alphabetical order. For alphabetical ordering, the lexicographical precedence is numeric values, uppercase letters, and lowercase letters. For count ordering, values with the same count are ordered alphabetically. With the [curl](#) command, URL-encode the [+](#) symbol as [%2B](#) .

Allowable values: [[alphabetical](#) , [count](#)]

Default: [alphabetical](#)

Response

Words

Information about the words

words	
Information about the words from a custom language model.	
words Always included *	An array of Word objects that provides information about each word in the custom model's words resource. The array is empty if the custom model has no words. > List[Word]
Status Code	
200	OK. The request succeeded.
400	Bad Request. The specified customization ID is invalid: <ul style="list-style-type: none">Malformed GUID: '{customization_id}'

Example responses

Success example

```
{
  "words": [
    {
      "word": "75.00",
      "sounds_like": [
        "75 dollars"
      ],
      "display_as":
"75.00",
      "count": 1,
      "source": [
        "user"
      ],
      "error": [
        {
          "75 dollars":
"Numbers are not allowed
in sounds_like. You can
try for example 'seventy
five dollars'."
        }
      ]
    },
    {
      "word": "HHonors",
      "sounds_like": [
        "hilton honors",
        "H. honors"
      ],
```

401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none"> Invalid customization_id '{customization_id}' for user
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

```

      "display_as":
"HHonors",
      "count": 1,
      "source": [
        "corpus1",
        "user"
      ]
    },
    {
      "words": [
        {
          "display_as":
"HHonors",
          "sounds_like": [
            "H. honors",
            "hilton
honors"
          ],
          "mapping_only":
true,
          "count": 1,
          "source": [
            "user"
          ],
          "word":
"HHonors"
        },
        {
          "display_as":
"IEEE",
          "sounds_like": [
            "I. triple E."
          ],
          "count": 1,

```

```
        "source": [
            "user"
        ],
        "word": "IEEE"
    }
]
},
{
    "word": "IEEE",
    "sounds_like": [
        "I. triple E."
    ],
    "display_as":
    "IEEE",
    "count": 3,
    "source": [
        "corpus1",
        "corpus2",
        "user"
    ]
},
{
    "word": "NCAA",
    "sounds_like": [
        "N. C. A. A.",
        "N. C. double A."
    ],
    "display_as":
    "NCAA",
    "count": 3,
    "source": [
        "corpus3",
        "user"
    ]
}
```



```

    },
    {
      "word": "tomato",
      "sounds_like": [
        "tomatoh",
        "tomayto"
      ],
      "display_as":
        "tomato",
      "count": 1,
      "source": [
        "user"
      ]
    }
  ]
}

```

Add custom words

Adds one or more custom words to a custom language model. You can use this method to add words or to modify existing words in a custom model's words resource. *For custom models that are based on previous-generation models*, the service populates the words resource for a custom model with out-of-vocabulary (OOV) words from each corpus or grammar that is added to the model. You can

```

add_words(
    self,
    customization_id:
      str,
    words:
      List['CustomWord'],
    **kwargs,
) -> DetailedResponse

```

use this method to modify OOV words in the model's words resource.

For a custom model that is based on a previous-generation model, the words resource for a model can contain a maximum of 90 thousand custom (OOV) words. This includes words that the service extracts from corpora and grammars and words that you add directly.

You must use credentials for the instance of the service that owns a model to add or modify custom words for the model. Adding or modifying custom words does not affect the custom model until you train the model for the new data by using the [Train a custom language model](#) method.

You add custom words by providing a `CustomWords` object, which is an array of `CustomWord` objects, one per word. Use the object's `word` parameter to identify the word that is to be added. You can also provide one or both of the optional `display_as` or `sounds_like` fields for each word.

- The `display_as` field provides a different way of spelling the word in a transcript. Use the parameter when you want the word to appear different from its usual representation or from its spelling in training data. For example, you might indicate that the word `IBM` is to be displayed as `IBM™` .

- The `sounds_like` field provides an array of one or more pronunciations for the word. Use the parameter to specify how the word can be pronounced by users. Use the parameter for words that are difficult to pronounce, foreign words, acronyms, and so on. For example, you might specify that the word `IEEE` can sound like `I triple E`. You can specify a maximum of five sounds-like pronunciations for a word. *For a custom model that is based on a previous-generation model*, if you omit the `sounds_like` field, the service attempts to set the field to its pronunciation of the word. It cannot generate a pronunciation for all words, so you must review the word's definition to ensure that it is complete and valid.

If you add a custom word that already exists in the words resource for the custom model, the new definition overwrites the existing data for the word. If the service encounters an error with the input data, it returns a failure code and does not add any of the words to the words resource.

The call returns an HTTP 201 response code if the input data is valid. It then asynchronously processes the words to add them to the model's words resource. The time that it takes for the analysis to complete depends on the number of new words that you add but is generally faster than adding a corpus or

grammar.

You can monitor the status of the request by using the [Get a custom language model](#) method to poll the model's status. Use a loop to check the status every 10 seconds. The method returns a `Customization` object that includes a `status` field. A status of `ready` means that the words have been added to the custom model. The service cannot accept requests to add new data or to train the model until the existing request completes.

You can use the [List custom words](#) or [Get a custom word](#) method to review the words that you add. Words with an invalid `sounds_like` field include an `error` field that describes the problem. You can use other words-related methods to correct errors, eliminate typos, and modify how words are pronounced as needed.

See also:

- [Add words to the custom language model](#)
- [Working with custom words for previous-generation models](#)
- [Working with custom words for next-generation models](#)
- [Validating a words resource for previous-generation models](#)
- [Validating a words resource for next-generation models](#) .

Request

parameters

customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

words

Required *

An array of [CustomWord](#) objects that provides information about each custom word that is to be added to or updated in the custom language model.

> [List\[CustomWord\]](#)

Example request for IBM Cloud

```
from ibm_watson import
SpeechToTextV1
from
ibm_watson.speech_to_text_
v1 import CustomWord
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

HHonors = CustomWord(
    'HHonors',
    ['hilton honors', 'H.
honors'],
```

```
'HHonors'
)
IEEE = CustomWord(
    'IEEE',
    ['I. triple E.']
)

speech_to_text.add_words(
    '{customization_id}',
    [HHonors, IEEE]
)

# Poll for language model
status.
```

Example request for IBM Cloud Pak for Data 

Response

Status Code

201 **Created.** Addition of the custom words was successfully started. The service is analyzing the data.

400 **Bad Request.** A required parameter is null or invalid, the JSON input is invalid, or the maximum number of sounds-like pronunciations for a word is exceeded. Specific failure messages include:

Example responses

Success example

```
{}
```



- Malformed GUID:
'{customization_id}'
 - Required property
'{property}' is missing
in JSON '{JSON}'
 - Word '{word}' contains
invalid character
character
 - Maximum number of
sounds-like for a word
exceeded
 - Maximum number of
allowed phones of one
item of sounds_like for
word '{word}' exceeded
 - Malformed JSON: '{JSON}'
 - Wrong type of parameter
'{parameter}' detected in
the passed JSON
 - TOTAL_NUMBER_OF_OOV_WORDS_EXCEEDS_MAXIMUM_ALLOWED
_FORMAT: "Total number of
OOV words {total_number}
exceeds
{maximum_allowed}"
-

401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none"> <code>Invalid customization_id '{customization_id}' for user</code>
409	Conflict. The service is currently busy handling a previous request for the custom model: <ul style="list-style-type: none"> <code>Customization '{customization_id}' is currently locked to process your last request.</code>
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Add a custom word

Adds a custom word to a custom language model. You can use this method to add a word or to modify an existing word in the words

```
add_word(
    self,
    customization id:
```



resource. *For custom models that are based on previous-generation models*, the service populates the words resource for a custom model with out-of-vocabulary (OOV) words from each corpus or grammar that is added to the model. You can use this method to modify OOV words in the model's words resource.

For a custom model that is based on a previous-generation models, the words resource for a model can contain a maximum of 90 thousand custom (OOV) words. This includes words that the service extracts from corpora and grammars and words that you add directly.

You must use credentials for the instance of the service that owns a model to add or modify a custom word for the model. Adding or modifying a custom word does not affect the custom model until you train the model for the new data by using the [Train a custom language model](#) method.

Use the `word_name` parameter to specify the custom word that is to be added or modified. Use the `CustomWord` object to provide one or both of the optional `display_as` or `sounds_like` fields for the word.

- The `display_as` field provides a different way of spelling the word in a transcript. Use the parameter when you want the word to appear different from its usual representation

```
str,
    word_name: str,
    *,
    word: str = None,
    sounds_like:
List[str] = None,
    display_as: str =
None,
    **kwargs,
) -> DetailedResponse
```

or from its spelling in training data. For example, you might indicate that the word `IBM` is to be displayed as `IBM™` .

- The `sounds_like` field provides an array of one or more pronunciations for the word. Use the parameter to specify how the word can be pronounced by users. Use the parameter for words that are difficult to pronounce, foreign words, acronyms, and so on. For example, you might specify that the word `IEEE` can sound like `i triple e` . You can specify a maximum of five sounds-like pronunciations for a word.
For custom models that are based on previous-generation models, if you omit the `sounds_like` field, the service attempts to set the field to its pronunciation of the word. It cannot generate a pronunciation for all words, so you must review the word's definition to ensure that it is complete and valid.

If you add a custom word that already exists in the words resource for the custom model, the new definition overwrites the existing data for the word. If the service encounters an error, it does not add the word to the words resource. Use the [Get a custom word](#) method to review the word that you add.

See also:

- [Add words to the custom language model](#)
- [Working with custom words for previous-](#)

[generation models](#)

- [Working with custom words for next-generation models](#)
- [Validating a words resource for previous-generation models](#)
- [Validating a words resource for next-generation models](#) .

Request

parameters

customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

Example request for IBM Cloud



word_name

Required *

str

The custom word that is to be added to or updated in the custom language model.

Do not use characters that need to be URL-encoded, for example, spaces, slashes, backslashes, colons, ampersands, double quotes, plus signs, equals signs, or question marks. Use a - (dash) or _ (underscore) to connect the tokens of compound words. URL-encode the word if it includes non-ASCII characters. For more information, see [Character encoding](#).

```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator
```

```
authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(
```

```
authenticator=authenticato
r
)
```

```
speech_to_text.set_service
_url('{url}')
```

```
speech_to_text.add_word(
    '{customization_id}',
    'NCAA',
    sounds_like=['N. C. A.
A.', 'N. C. double A.'],
    display_as='NCAA'
)
```

Example request for IBM Cloud Pak for Data

word

str

For the [Add custom words](#) method, you must specify the custom word that is to be added to or updated in the custom model. Do not use characters that need to be URL-encoded, for example, spaces, slashes, backslashes, colons, ampersands, double quotes, plus signs, equals signs, or question marks. Use a - (dash) or _ (underscore) to connect the tokens of compound words. A Japanese custom word can include at most 25 characters, not including leading or trailing spaces. Omit this parameter for the [Add a custom word](#) method.

sounds_like

List[str]

As array of sounds-like pronunciations for the custom word. Specify how words that are difficult to pronounce, foreign words, acronyms, and so on can be pronounced by users.

- *For custom models that*

are based on previous-generation models, for a word that is not in the service's base vocabulary, omit the parameter to have the service automatically generate a sounds-like pronunciation for the word.

- For a word that is in the service's base vocabulary, use the parameter to specify additional pronunciations for the word. You cannot override the default pronunciation of a word; pronunciations you add augment the pronunciation from the base vocabulary.

A word can have at most five sounds-like pronunciations. A pronunciation can include at most 40 characters, not including leading or trailing spaces. A Japanese pronunciation can include at most 25 characters, not including leading or trailing spaces.

display_as

str

An alternative spelling for the custom word when it appears in a transcript. Use the parameter when you want the word to have a spelling that is different from its usual representation or from its spelling in corpora training data.

For custom models that are based on next-generation models, the service uses the spelling of the word as the display-as value if you omit the field.

Response

Status Code

201

Created. The custom word was successfully added to the custom language model.

Example responses

Success example

```
{ }
```

400

Bad Request. The specified customization ID is invalid, or the maximum number of sounds-like pronunciations for a word is exceeded. Specific failure messages include:

- `Malformed GUID:`
`'{customization_id}'`
- `Maximum number of sounds-like for a word exceeded`
- `Maximum number of allowed phones of one item of sounds_like for word '{word}' exceeded`
- `TOTAL_NUMBER_OF_OOV_WORDS_EXCEEDS_MAXIMUM_ALLOWED`
`_FORMAT: "Total number of OOV words {total_number} exceeds {maximum_allowed}"`

401

Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials:

- `Invalid customization_id`
`'{customization_id}' for user`
-

405	Method Not Allowed. The word name includes characters that need to be URL-encoded.
409	Conflict. The service is currently busy handling a previous request for the custom model: <ul style="list-style-type: none">Customization '{customization_id}' is currently locked to process your last request.
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Get a custom word

Gets information about a custom word from a custom language model. You must use credentials for the instance of the service that owns a model to list information about its words.

See also: [Listing words from a custom language model](#) .

Request

parameters

customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

```
get_word(  
    self,  
    customization_id:  
    str,  
    word_name: str,  
    **kwargs,  
) -> DetailedResponse
```



Example request for IBM Cloud



word_name

Required *

str

The custom word that is to be read from the custom language model. URL-encode the word if it includes non-ASCII characters. For more information, see [Character encoding](#).

```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

word =
speech_to_text.get_word(
    '{customization_id}',
    'NCAA'
).get_result()
print(json.dumps(word,
indent=2))
```

Example request for IBM Cloud Pak for Data ▾

Response

Word	Information about a word from a custom language model.
word Always included * str	A word from the custom model's words resource. The spelling of the word is used to train the model.

Example responses

Success example

```
{
  "word": "NCAA",
  "sounds_like": [
    "N. C. A. A.",
    "N. C. double A."
  ],
  "display_as": "NCAA",
  "count": 3,
  "source": [
    "corpus3",
    "user"
  ]
}
```

sounds_like

Always included *
List[str]

An array of as many as five pronunciations for the word.

- *For a custom model that is based on a previous-generation model*, in addition to sounds-like pronunciations that were added by a user, the array can include a sounds-like pronunciation that is automatically generated by the service if none is provided when the word is added to the custom model.
 - *For a custom model that is based on a next-generation model*, the array can include only sounds-like pronunciations that were added by a user.
-

display_as

Always included *
str

The spelling of the word that the service uses to display the word in a transcript.

- *For a custom model that is based on a previous-generation model*, the field can contain an empty string if no display-as value is provided for a word that exists in the service's base vocabulary. In this case, the word is displayed as it is spelled.
 - *For a custom model that is based on a next-generation model*, the service uses the spelling of the word as the value of the display-as field when the word is added to the model.
-

count

Always included *
int

For a custom model that is based on a previous-generation model, a sum of the number of times the word is found across all corpora and grammars. For example, if the word occurs five times in one corpus and seven times in another, its count is **12** . If you add a custom word to a model before it is added by any corpora or grammars, the count begins at **1** ; if the word is added from a corpus or grammar first and later modified, the count reflects only the number of times it is found in corpora and grammars.

For a custom model that is based on a next-generation model, the **count** field for any word is always **1** .

source

Always included *
List[str]

An array of sources that describes how the word was added to the custom model's words resource.

- *For a custom model that is*

based on previous-generation model, the field includes the name of each corpus and grammar from which the service extracted the word. For OOV that are added by multiple corpora or grammars, the names of all corpora and grammars are listed. If you modified or added the word directly, the field includes the string `user`.

- *For a custom model that is based on a next-generation model*, this field shows only `user` for custom words that were added directly to the custom model. Words from corpora and grammars are not added to the words resource for custom models that are based on next-generation models.
-

error

If the service discovered one or more problems that you need to correct for the word's definition, an array that describes each of the errors.

> [List\[WordError\]](#)

Status Code

200

OK. The request succeeded.

400

Bad Request. The specified customization ID or word is invalid, including the case where the word does not exist for the custom model. Specific failure messages include:

- `Malformed GUID: '{customization_id}'`
- `Invalid value for word '{word}'`

401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none">• <code>Invalid customization_id '{customization_id}' for user</code>
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.


Delete a custom word

Deletes a custom word from a custom language model. You can remove any word that you added to the custom model's words resource via any means. However, if the word also exists in the service's base vocabulary, the service removes the word only from the words resource; the word remains in the base vocabulary. Removing a custom word does not affect the custom model until you train the model with the [Train a custom language model](#) method. You must use credentials for the instance of the service that owns a model to delete its words.

See also: [Deleting a word from a custom language model](#) .

Request

parameters



```
delete_word(  
    self,  
    customization_id:  
    str,  
    word_name: str,  
    **kwargs,  
) -> DetailedResponse
```

Example request for IBM Cloud



customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

word_name

Required *

str

The custom word that is to be deleted from the custom language model. URL-encode the word if it includes non-ASCII characters. For more information, see [Character encoding](#).

```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator
```

```
authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(
```

```
authenticator=authenticato
r
)
```

```
speech_to_text.set_service
_url('{url}')
```

```
speech_to_text.delete_word
(
    '{customization_id}',
    'NCAA'
)
```

Example request for IBM Cloud Pak for Data

Response

Status Code

Status code

200	OK. The custom word was successfully deleted from the custom language model.
400	Bad Request. The specified customization ID or word is invalid, including the case where the word does not exist for the custom model. Specific failure messages include: <ul style="list-style-type: none">Malformed GUID: <code>'{customization_id}'</code>Invalid value for word <code>'{word}'</code>
401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none">Invalid customization_id <code>'{customization_id}'</code> for user
405	Method Not Allowed. No word name was specified with the request.

Example responses

Success example

```
{}
```

409	Conflict. The service is currently busy handling a previous request for the custom model: <ul style="list-style-type: none">• Customization '{customization_id}' is currently locked to process your last request.
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

List grammars

Lists information about all grammars from a custom language model. For each grammar, the information includes the name, status, and (for grammars that are based on previous-generation models) the total number of out-of-vocabulary (OOV) words. You must use credentials for the instance of the service that owns a model to list its grammars.

See also:

- [Listing grammars from a custom language model](#)
- [Language support for customization](#) .

Request

parameters

customization_id

Required *

str


The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

```
list_grammars(  
    self,  
    customization_id:  
str,  
    **kwargs,  
) -> DetailedResponse
```



Example request for IBM Cloud





```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

grammars =
speech_to_text.list_gramma
rs('{customization_id}').g
et_result()
print(json.dumps(grammars,
indent=2))
```

Example request for IBM Cloud Pak for Data 

Response

Grammars

Information about the grammars from a custom language model.

Example responses

grammars

Always included *

An array of [Grammar](#) objects that provides information about the grammars for the custom model. The array is empty if the custom model has no grammars.

> [List\[Grammar\]](#)

Success example

Status Code

200

OK. The request succeeded.

400

Bad Request. The specified customization ID is invalid:

- [Malformed GUID](#):
`'{customization_id}'`

401 Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials:

- `Invalid customization_id '{customization_id}' for user`

500 Internal Server Error. The service experienced an internal error.

503 Service Unavailable. The service is currently unavailable.

```
{
  "grammars": [
    {
      "out_of_vocabulary_words":
0,
      "name": "confirm-
xml",
      "status": "analyzed"
    },
    {
      "out_of_vocabulary_words":
0,
      "name": "confirm-
abnf",
      "status": "analyzed"
    },
    {
      "out_of_vocabulary_words":
8,
      "name": "list-abnf",
      "status": "analyzed"
    }
  ]
}
```

Add a grammar

Adds a single grammar file to a custom language model. Submit a plain text file in UTF-8 format that defines the grammar. Use multiple requests to submit multiple grammar files. You must use credentials for the instance of the service that owns a model to add a grammar to it. Adding a grammar does not affect the custom language model until you train the model for the new data by using the [Train a custom language model](#) method.

The call returns an HTTP 201 response code if the grammar is valid. The service then asynchronously processes the contents of the grammar and automatically extracts new words that it finds. This operation can take a few seconds or minutes to complete depending on the size and complexity of the grammar, as well as the current load on the service. You cannot submit requests to add additional resources to the custom model or to train the model until the service's analysis of the grammar for the current request completes. Use the [Get a grammar](#) method to check the status of the analysis.

For grammars that are based on previous-

```
add_grammar(  
    self,  
    customization_id:  
    str,  
    grammar_name: str,  
    grammar_file:  
    BinaryIO,  
    content_type: str,  
    *,  
    allow_overwrite:  
    bool = None,  
    **kwargs,  
) -> DetailedResponse
```

generation models, the service populates the model's words resource with any word that is recognized by the grammar that is not found in the model's base vocabulary. These are referred to as out-of-vocabulary (OOV) words. You can use the [List custom words](#) method to examine the words resource and use other words-related methods to eliminate typos and modify how words are pronounced as needed. *For grammars that are based on next-generation models*, the service extracts no OOV words from the grammars.

To add a grammar that has the same name as an existing grammar, set the `allow_overwrite` parameter to `true` ; otherwise, the request fails. Overwriting an existing grammar causes the service to process the grammar file and extract OOV words anew. Before doing so, it removes any OOV words associated with the existing grammar from the model's words resource unless they were also added by another resource or they have been modified in some way with the [Add custom words](#) or [Add a custom word](#) method.

For grammars that are based on previous-generation models, the service limits the overall amount of data that you can add to a custom model to a maximum of 10 million total words from all sources combined. Also, you can add no more than 90 thousand OOV

words to a model. This includes words that the service extracts from corpora and grammars and words that you add directly.

See also:

- [Understanding grammars](#)
- [Add a grammar to the custom language model](#)
- [Language support for customization](#) .

Request

parameters	
customization_id	
Required *	
str	
The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.	
grammar_name	The name of the new
Required *	grammar for the custom
str	

Example request for IBM Cloud

```
from os.path import join,
dirname
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(
```

language model. Use a localized name that matches the language of the custom model and reflects the contents of the grammar.

- Include a maximum of 128 characters in the name.
- Do not use characters that need to be URL-encoded. For example, do not use spaces, slashes, backslashes, colons, ampersands, double quotes, plus signs, equals signs, questions marks, and so on in the name. (The service does not prevent the use of these characters. But because they must be URL-encoded wherever used, their use is strongly discouraged.)
- Do not use the name of an existing grammar or corpus that is already defined for the custom model.
- Do not use the name

```
authenticator=authenticato  
r  
)
```

```
speech_to_text.set_service  
_url('{url}')
```

```
with  
open(join(dirname(__file__  
) , './.', 'list.abnf'),  
      'rb') as
```

```
grammar_file:
```

```
speech_to_text.add_grammar  
(
```

```
'{customization_id}',  
  'list-abnf',  
  grammar_file,  
  'application/srgs'  
)
```

```
# Poll for grammar status.
```

Download sample file [list.abnf](#)

Example request for IBM Cloud Pak for Data 

`user` , which is reserved by the service to denote custom words that are added or modified by the user.

- Do not use the name `base_lm` or `default_lm` . Both names are reserved for future use by the service.

grammar_file

Required *

BinaryIO

A plain text file that contains the grammar in the format specified by the `Content-Type` header. Encode the file in UTF-8 (ASCII is a subset of UTF-8). Using any other encoding can lead to issues when compiling the grammar or to unexpected results in decoding. The service ignores an encoding that is specified in the header of the grammar.

With the `curl` command, use the `--data-binary` option to upload the file for the request.

content_type

Required *

str

The format (MIME type) of the grammar file:

- `application/srgs` for Augmented Backus-Naur Form (ABNF), which uses a plain-text representation that is similar to traditional BNF grammars.
- `application/srgs+xml` for XML Form, which uses XML elements to represent the grammar.

Allowable values: [

`application/srgs` ,
`application/srgs+xml`]

allow_overwrite If `true` , the specified

bool

grammar overwrites an existing grammar with the same name. If `false` , the request fails if a grammar with the same name already exists. The parameter has no effect if a grammar with the same name does not already exist.

Default: `false`

Response

Status Code	
201	Created. Addition of the grammar was successfully started. The service is analyzing the data.
400	Bad Request. A required parameter is null or invalid, the specified grammar name already exists, or the custom model needs to be upgraded, among other possibilities. Specific failure messages include: <ul style="list-style-type: none">Malformed GUID: <code>'{customization_id}'</code>Grammar file not specified or emptyGrammar <code>'{grammar_name}'</code> already exists - change its name, remove existing grammar before adding new one, or overwrite existing grammar by setting <code>'allow_overwrite'</code> to <code>'true'</code>Corpus exists with grammar name <code>'{grammar_name}'</code>. Please

Example responses

Success example

`{}`

use different name.

- TOTAL_NUMBER_OF_OOV_WORDS_EXCEEDS_MAXIMUM_ALLOWED_FORMAT: "Total number of OOV words {total_number} exceeds {maximum_allowed}"
 - Analysis of grammar '{grammar_name}' failed due to {error_message}. Please fix the error then add the grammar again by setting the 'allow_overwrite' flag to 'true'. , where {error_message} is a message of the form {"code": 404, "error": "Model en-US_BroadbandModel (version: en-US_BroadbandModel. {version}) not found", "code_description": "Not Found"} . Upgrade the custom language model to the latest version of its base language model, and then add the grammar to the custom model.
-

401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none">• <code>Invalid customization_id '{customization_id}' for user</code>
405	Method Not Allowed. The grammar name includes characters that need to be URL-encoded.
409	Conflict. The service is currently busy handling a previous request for the custom model: <ul style="list-style-type: none">• <code>Customization '{customization_id}' is currently locked to process your last request.</code>
415	Unsupported Media Type. The request specified an unacceptable media type.

500	Internal Server Error. An internal error prevented the service from satisfying the request. You can also receive status code 500 Forwarding Error if the service is currently busy handling a previous request for the custom model.
503	Service Unavailable. The service is currently unavailable.

Get a grammar

Gets information about a grammar from a custom language model. For each grammar, the information includes the name, status, and (for grammars that are based on previous-generation models) the total number of out-of-vocabulary (OOV) words. You must use credentials for the instance of the service that owns a model to list its grammars.

See also:

- [Listing grammars from a custom language model](#)
- [Language support for customization](#) .

Request

```
get_grammar(  
    self,  
    customization_id:  
    str,  
    grammar_name: str,  
    **kwargs,  
) -> DetailedResponse
```



parameters

customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.


grammar_name The name of the grammar for

Required * the custom language model.

str

Example request for IBM Cloud





```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

grammar =
speech_to_text.get_grammar
(
    '{customization_id}',
    'list-abnf'
).get_result()
print(json.dumps(grammar,
indent=2))
```

Example request for IBM Cloud Pak for Data 

Response

Grammar	Information about a grammar from a custom language model.
name Always included * str	The name of the grammar.
out_of_vocabulary_words Always included * int	<p><i>For custom models that are based on previous-generation models, the number of OOV words extracted from the grammar. The value is 0 while the grammar is being processed.</i></p> <p><i>For custom models that are based on next-generation models, no OOV words are extracted from grammars, so the value is always 0 .</i></p>

Example responses

Success example

```
{  
  
  "out_of_vocabulary_words":  
    8,  
  "name": "list-abnf",  
  "status": "analyzed"  
}
```

status

Always included *
str

The status of the grammar:

- **analyzed** : The service successfully analyzed the grammar. The custom model can be trained with data from the grammar.
- **being_processed** : The service is still analyzing the grammar. The service cannot accept requests to add new resources or to train the custom model.
- **undetermined** : The service encountered an error while processing the grammar. The **error** field describes the failure.

Possible values: [

analyzed ,
being_processed ,
undetermined]

error

str

If the status of the grammar is `undetermined`, the following message:

```
Analysis of grammar
'{grammar_name}'
failed. Please try
fixing the error or
adding the grammar
again by setting the
'allow_overwrite' flag
to 'true'..
```

Status Code

200

OK. The request succeeded.

400

Bad Request. The specified customization ID or grammar name is invalid, including the case where the grammar does not exist for the custom model. Specific failure messages include:

- `Malformed GUID:`
`'{customization_id}'`
- `Invalid value for`
`grammar name`
`'{grammar_name}'`

401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none">• <code>Invalid customization_id '{customization_id}' for user</code>
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Delete a grammar


Deletes an existing grammar from a custom language model. *For grammars that are based on previous-generation models*, the service removes any out-of-vocabulary (OOV) words associated with the grammar from the custom model's words resource unless they were also added by another resource or they were modified in some way with the [Add custom words](#) or [Add a custom word](#) method. Removing a grammar does not affect the custom model until you train the model with the [Train a custom language model](#) method. You must use credentials for the instance of the service that owns a model to delete its grammar.

See also:

- [Deleting a grammar from a custom language model](#)
- [Language support for customization](#) .

Request

parameters



```
delete_grammar(  
    self,  
    customization_id:  
    str,  
    grammar_name: str,  
    **kwargs,  
) -> DetailedResponse
```

Example request for IBM Cloud



customization_id

Required *

str

The customization ID (GUID) of the custom language model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

grammar_name The name of the grammar for

Required * the custom language model.

str

```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator
```

```
authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(
```

```
authenticator=authenticato
r
)
```

```
speech_to_text.set_service
_url('{url}')
```

```
speech_to_text.delete_gram
mar(
    '{customization_id}',
    'list-abnf'
)
```

Example request for IBM Cloud Pak for Data

Response

Status Code

Status code		Example responses
200	OK. The grammar was successfully deleted from the custom language model.	<div>Success example</div> <div><pre>{}</pre></div>
400	Bad Request. The specified customization ID or grammar name is invalid, including the case where the grammar does not exist for the custom model. Specific failure messages include: <ul style="list-style-type: none">Malformed GUID: <code>'{customization_id}'</code>Invalid value for grammar name <code>'{grammar_name}'</code>	
401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none">Invalid customization_id <code>'{customization_id}'</code> for user	
405	Method Not Allowed. No grammar name was specified with the request.	

409

Conflict. The service is currently busy handling a previous request for the custom model:

- Customization
'{customization_id}' is currently locked to process your last request.

500

Internal Server Error. The service experienced an internal error.

503

Service Unavailable. The service is currently unavailable.

Create a custom
acoustic model

Creates a new custom acoustic model for a specified base model. The custom acoustic model can be used only with the base model for which it is created. The model is owned by the instance of the service whose credentials are used to create it.

You can create a maximum of 1024 custom acoustic models per owning credentials. The service returns an error if you attempt to create more than 1024 models. You do not lose any models, but you cannot create any more until your model count is below the limit.

Note: Acoustic model customization is supported only for use with previous-generation models. It is not supported for next-generation models.

Important: Effective **31 July 2023**, all previous-generation models will be removed from the service and the documentation. Most previous-generation models were deprecated on 15 March 2022. You must migrate to the equivalent next-generation model by 31 July 2023. For more information, see [Migrating to next-generation models](#) .

See also: [Create a custom acoustic model](#) .

Request

parameters

```
create_acoustic_model(  
    self,  
    name: str,  
    base_model_name:  
    str,  
    *,  
    description: str =  
    None,  
    **kwargs,  
) -> DetailedResponse
```



name

Required *

str

A user-defined name for the new custom acoustic model. Use a localized name that matches the language of the custom model. Use a name that describes the acoustic environment of the custom model, such as `Mobile custom model` or `Noisy car custom model`. Use a name that is unique among all custom acoustic models that you own.

Include a maximum of 256 characters in the name. Do not use backslashes, slashes, colons, equal signs, ampersands, or question marks in the name.

base_model_name

Required *

str

The name of the base language model that is to be customized by the new custom acoustic model. The new custom model can be used only with the base

Example request for IBM Cloud

```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

acoustic_model =
speech_to_text.create_acou
stic_model(
    'First example
acoustic model',
    'en-
US_BroadbandModel',
    description='First
custom acoustic model
example'
```


model that it customizes.

To determine whether a base model supports acoustic model customization, refer to [Language support for customization](#).

Allowable values: [

ar-MS_BroadbandModel ,
de-DE_BroadbandModel ,
de-DE_NarrowbandModel
, en-AU_BroadbandModel ,
en-AU_NarrowbandModel
, en-GB_BroadbandModel ,
en-GB_NarrowbandModel
, en-US_BroadbandModel ,
en-US_NarrowbandModel
,
en-
US_ShortForm_NarrowbandModel
, es-AR_BroadbandModel ,
es-AR_NarrowbandModel
, es-CL_BroadbandModel ,
es-CL_NarrowbandModel
, es-CO_BroadbandModel ,
es-CO_NarrowbandModel
, es-ES_BroadbandModel ,
es-ES_NarrowbandModel

```
).get_result()  
print(json.dumps(acoustic_  
model, indent=2))
```

Example request for IBM Cloud Pak for Data 

```
, es-MX_BroadbandModel ,  
  es-MX_NarrowbandModel  
, es-PE_BroadbandModel ,  
  es-PE_NarrowbandModel  
, fr-CA_BroadbandModel ,  
  fr-CA_NarrowbandModel  
, fr-FR_BroadbandModel ,  
  fr-FR_NarrowbandModel  
, it-IT_BroadbandModel ,  
  it-IT_NarrowbandModel  
, ja-JP_BroadbandModel ,  
  ja-JP_NarrowbandModel  
, ko-KR_BroadbandModel ,  
  ko-KR_NarrowbandModel  
, nl-NL_BroadbandModel ,  
  nl-NL_NarrowbandModel  
, pt-BR_BroadbandModel ,  
  pt-BR_NarrowbandModel  
, zh-CN_BroadbandModel ,  
  zh-CN_NarrowbandModel  
]
```

description

str

A recommended description of the new custom acoustic model. Use a localized description that matches the language of the custom model. Include a maximum of 128 characters in the description.

Response

AcousticModel

Information about an existing custom acoustic model.

customization_id

Always included *

str

The customization ID (GUID) of the custom acoustic model. The [Create a custom acoustic model](#) method returns only this field of the object; it does not return the other fields.

Example responses

Success example

```
{
  "customization_id":
  "74f4807e-b5ff-4866-824e-
  6bba1a84fe96"
}
```

created

str

The date and time in Coordinated Universal Time (UTC) at which the custom acoustic model was created. The value is provided in full ISO 8601 format (`YYYY-MM-DDThh:mm:ss.sTZD`).

updated

str

The date and time in Coordinated Universal Time (UTC) at which the custom acoustic model was last modified. The `created` and `updated` fields are equal when an acoustic model is first added but has yet to be updated. The value is provided in full ISO 8601 format (YYYY-MM-DDThh:mm:ss.sTZD).

language

str

The language identifier of the custom acoustic model (for example, `en-US`).

versions List[str]	A list of the available versions of the custom acoustic model. Each element of the array indicates a version of the base model with which the custom model can be used. Multiple versions exist only if the custom model has been upgraded to a new version of its base model. Otherwise, only a single version is shown.
------------------------------	---

owner str	The GUID of the credentials for the instance of the service that owns the custom acoustic model.
---------------------	--

name str	The name of the custom acoustic model.
--------------------	--

description str	The description of the custom acoustic model.
---------------------------	---

base_model_name str	The name of the language model for which the custom acoustic model was created.
-------------------------------	---

status

str

The current status of the custom acoustic model:

- **pending** : The model was created but is waiting either for valid training data to be added or for the service to finish analyzing added data.
- **ready** : The model contains valid data and is ready to be trained. If the model contains a mix of valid and invalid resources, you need to set the **strict** parameter to **false** for the training to proceed.
- **training** : The model is currently being trained.
- **available** : The model is trained and ready to use.
- **upgrading** : The model is currently being upgraded.
- **failed** : Training of the model failed.

Possible values: [**pending** ,

`ready , training ,
available , upgrading ,
failed]`

progress

int

A percentage that indicates the progress of the custom acoustic model's current training. A value of `100` means that the model is fully trained. **Note:** The `progress` field does not currently reflect the progress of the training. The field changes from `0` to `100` when training is complete.

warnings

str

If the request included unknown parameters, the following message:
`Unexpected query
parameter(s)
['parameters']
detected` , where `parameters` is a list that includes a quoted string for each unknown parameter.

Status Code

201

Created. The custom acoustic model was successfully created.

400

Bad Request. A required parameter is null or invalid. Specific failure messages include:

- Required parameter '{name}' is missing
 - Required parameter '{name}' cannot be empty string
 - Required parameter '{name}' cannot be null
 - The base model '{model_name}' is not recognized
 - Acoustic customization is not supported for base model '{model_name}'
 - You exceeded the maximum '{model_number}' of allowed custom acoustic models. You have '{model_number}' custom acoustic models. Please remove the models you do not need or contact the IBM speech support team to apply for an exception.
-

401	Unauthorized. The specified credentials are invalid.
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

List custom acoustic models

Lists information about all custom acoustic models that are owned by an instance of the service. Use the `language` parameter to see all custom acoustic models for the specified language. Omit the parameter to see all custom acoustic models for all languages. You must use credentials for the instance of the service that owns a model to list information about it.

Note: Acoustic model customization is supported only for use with previous-generation models. It is not supported for next-generation models.

See also: [Listing custom acoustic models](#) .

```
list_acoustic_models(  
    self,  
    *,  
    language: str =  
None,  
    **kwargs,  
) -> DetailedResponse
```

Request

parameters

language
str

The identifier of the language for which custom language or custom acoustic models are to be returned. Specify the five-character language identifier; for example, specify **en-US** to see all custom language or custom acoustic models that are based on US English models. Omit the parameter to see all custom language or custom acoustic models that are owned by the requesting credentials.

To determine the languages for which customization is available, see [Language support for customization](#).

Allowable values: [**ar-MS** , **cs-CZ** , **de-DE** , **en-AU** , **en-GB** , **en-IN** , **en-US** , **en-WW** , **es-AR** , **es-CL** ,

Example request for IBM Cloud

```
es-CO , es-ES , es-LA ,  
es-MX , es-PE , fr-CA ,  
fr-FR , hi-IN , it-IT ,  
ja-JP , ko-KR , nl-BE ,  
nl-NL , pt-BR , sv-SE ,  
zh-CN ]
```

```
import json  
from ibm_watson import  
SpeechToTextV1  
from  
ibm_cloud_sdk_core.authent  
icators import  
IAMAuthenticator  
  
authenticator =  
IAMAuthenticator('{apikey}  
' )  
speech_to_text =  
SpeechToTextV1(  
  
authenticator=authenticato  
r  
)  
  
speech_to_text.set_service  
_url('{url}')  
  
acoustic_models =  
speech_to_text.list_acoust  
ic_models('en-  
US').get_result()  
print(json.dumps(acoustic_  
models, indent=2))
```

Example request for IBM Cloud Pak for Data 

Response

AcousticModels Information about existing custom acoustic models.

customizations An array of Always included * **AcousticModel** objects that provides information about each available custom acoustic model. The array is empty if the requesting credentials own no custom acoustic models (if no language is specified) or own no custom acoustic models for the specified language.

> [List\[AcousticModel\]](#)

Status Code

200 OK. The request succeeded.

400 **Bad Request.** A required parameter is null or invalid. Specific failure messages include:

- `Language '{language}' is not supported for customization`

Example responses

Success example

```
{
  "customizations": [
    {
      "customization_id":
"74f4807e-b5ff-4866-824e-6bba1a84fa97",
      "created": "2016-06-01T18:42:25.324Z",
      "updated": "2020-01-19T11:12:02.296Z",
      "language": "en-US",
      "versions": [
        "en-US_BroadbandModel.v2018-07-31",
        "en-US_BroadbandModel.v2020-01-16"
      ],
      "owner": "297cfd08-330a-22ba-93ce-1a73f454dd98",
      "name": "Example model one",
      "description": "Example custom acoustic"
```

401	Unauthorized. The specified credentials are invalid.
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

```
model",
  "base_model_name":
"en-US_BroadbandModel",
  "status": "pending",
  "progress": 0
},
{
  "customization_id":
"8391f918-3b76-e109-763c-
b7732faa3312",
  "created": "2017-12-
01T18:51:37.291Z",
  "updated": "2017-12-
02T19:21:06.825Z",
  "language": "en-US",
  "versions": [
    "en-
US_BroadbandModel.v2017-
11-15"
  ],
  "owner": "297cfd08-
330a-22ba-93ce-
1a73f454dd98",
  "name": "Example
model two",
  "description":
"Example custom acoustic
model two",
  "base_model_name":
"en-US_BroadbandModel",
  "status":
"available",
  "progress": 100
}
```

Get a custom acoustic model

Gets information about a specified custom acoustic model. You must use credentials for the instance of the service that owns a model to list information about it.

Note: Acoustic model customization is supported only for use with previous-generation models. It is not supported for next-generation models.

See also: [Listing custom acoustic models](#) .

Request

parameters

```
]
}
```

```
get_acoustic_model(
    self,
    customization_id:
str,
    **kwargs,
) -> DetailedResponse
```

Example request for IBM Cloud

customization_id

Required *

str

The customization ID (GUID) of the custom acoustic model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

acoustic_model =
speech_to_text.get_acousti
c_model('{customization_id
}').get_result()
print(json.dumps(acoustic_
model, indent=2))
```

Example request for IBM Cloud Pak for Data

Response

AcousticModel Information about an existing custom acoustic model.

Example responses

Success example

customization_id

Always included *

str

The customization ID (GUID) of the custom acoustic model. The [Create a custom acoustic model](#) method returns only this field of the object; it does not return the other fields.

created

str

The date and time in Coordinated Universal Time (UTC) at which the custom acoustic model was created. The value is provided in full ISO 8601 format (**YYYY-MM-DDThh:mm:ss.sTZD**).

updated

str

The date and time in Coordinated Universal Time (UTC) at which the custom acoustic model was last modified. The **created** and **updated** fields are equal when an acoustic model is first added but has yet to be updated. The value is provided in full ISO 8601 format (YYYY-MM-DDThh:mm:ss.sTZD).

language

str

The language identifier of the custom acoustic model (for example, **en-US**).

versions

List[str]

A list of the available versions of the custom acoustic model. Each element of the array indicates a version of the base model with which the custom model can be used. Multiple versions exist only if the custom model has been upgraded to a new version of its base model. Otherwise, only a single version is shown.

```
{
  "customization_id":
  "74f4807e-b5ff-4866-824e-
  6bba1a84fa97",
  "created": "2016-06-
  01T18:42:25.324Z",
  "updated": "2020-01-
  19T11:12:02.296Z",
  "language": "en-US",
  "versions": [
    "en-
    US_BroadbandModel.v2018-
    07-31",
    "en-
    US_BroadbandModel.v2020-
    01-16"
  ],
  "owner": "297cfd08-330a-
  22ba-93ce-1a73f454dd98",
  "name": "Example model
  one",
  "description": "Example
  custom acoustic model",
  "base_model_name": "en-
  US_BroadbandModel",
  "status": "pending",
  "progress": 0
}
```



owner str	The GUID of the credentials for the instance of the service that owns the custom acoustic model.
---------------------	--

name str	The name of the custom acoustic model.
--------------------	--

description str	The description of the custom acoustic model.
---------------------------	---

base_model_name str	The name of the language model for which the custom acoustic model was created.
-------------------------------	---

status str	<p>The current status of the custom acoustic model:</p> <ul style="list-style-type: none">• pending : The model was created but is waiting either for valid training data to be added or for the service to finish analyzing added data.• ready : The model contains valid data and is ready to be trained. If the model contains a mix of valid and invalid
----------------------	---

resources, you need to set the `strict` parameter to `false` for the training to proceed.

- `training` : The model is currently being trained.
- `available` : The model is trained and ready to use.
- `upgrading` : The model is currently being upgraded.
- `failed` : Training of the model failed.

Possible values: [`pending` , `ready` , `training` , `available` , `upgrading` , `failed`]

progress

int

A percentage that indicates the progress of the custom acoustic model's current training. A value of `100` means that the model is fully trained. **Note:** The `progress` field does not currently reflect the progress of the training. The field changes from `0` to `100` when training is complete.

warnings

str

If the request included unknown parameters, the following message:

```
Unexpected query
parameter(s)
['parameters']
detected , where
parameters is a list that
includes a quoted string for
each unknown parameter.
```

Status Code

200

OK. The request succeeded.

400	Bad Request. The specified customization ID is invalid: <ul style="list-style-type: none">• <code>Malformed GUID:</code> <code>'{customization_id}'</code>
401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none">• <code>Invalid customization_id</code> <code>'{customization_id}'</code> for <code>user</code>
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Delete a custom acoustic model

Deletes an existing custom acoustic model. The custom model cannot be deleted if another request, such as adding an audio resource to the model, is currently being processed. You must use credentials for the instance of the service that owns a model to delete it.

Note: Acoustic model customization is supported only for use with previous-generation models. It is not supported for next-generation models.

See also: [Deleting a custom acoustic model](#).

Request

parameters

```
delete_acoustic_model(  
    self,  
    customization_id:  
str,  
    **kwargs,  
) -> DetailedResponse
```



Example request for IBM Cloud




customization_id

Required *

str

The customization ID (GUID) of the custom acoustic model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.



```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

speech_to_text.delete_acou
stic_model('{customization
_id}')
```

Example request for IBM Cloud Pak for Data 

Response

Status Code

Example responses

200	OK. The custom acoustic model was successfully deleted.
400	Bad Request. The specified customization ID is invalid: <ul style="list-style-type: none">Malformed GUID: <code>'{customization_id}'</code>
401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials, including the case where the custom model does not exist: <ul style="list-style-type: none">Invalid customization_id <code>'{customization_id}'</code> for user
409	Conflict. The service is currently busy handling a previous request for the custom model: <ul style="list-style-type: none">Customization <code>'{customization_id}'</code> is currently locked to process your last request.
500	Internal Server Error. The service experienced an internal error.

Success example

```
{}
```

Train a custom acoustic model

Initiates the training of a custom acoustic model with new or changed audio resources. After adding or deleting audio resources for a custom acoustic model, use this method to begin the actual training of the model on the latest audio data. The custom acoustic model does not reflect its changed data until you train it. You must use credentials for the instance of the service that owns a model to train it.

The training method is asynchronous. Training time depends on the cumulative amount of audio data that the custom acoustic model contains and the current load on the service. When you train or retrain a model, the service uses all of the model's audio data in the training. Training a custom acoustic model takes approximately as long as the length of its cumulative audio data. For example, it takes approximately 2 hours to train a model that contains a total of 2 hours of audio. The method returns an HTTP 200 response code to indicate that the training process has begun.

```
train_acoustic_model(  
    self,  
    customization_id:  
    str,  
    *,  
  
    custom_language_model_id:  
    str = None,  
    strict: bool =  
    None,  
    **kwargs,  
) -> DetailedResponse
```

You can monitor the status of the training by using the [Get a custom acoustic model](#) method to poll the model's status. Use a loop to check the status once a minute. The method returns an `AcousticModel` object that includes `status` and `progress` fields. A status of `available` indicates that the custom model is trained and ready to use. The service cannot train a model while it is handling another request for the model. The service cannot accept subsequent training requests, or requests to add new audio resources, until the existing training request completes.

You can use the optional `custom_language_model_id` parameter to specify the GUID of a separately created custom language model that is to be used during training. Train with a custom language model if you have verbatim transcriptions of the audio files that you have added to the custom model or you have either corpora (text files) or a list of words that are relevant to the contents of the audio files. For training to succeed, both of the custom models must be based on the same version of the same base model, and the custom language model must be fully trained and available.

Note: Acoustic model customization is supported only for use with previous-generation models. It is not supported for next-generation models.

See also:

- [Train the custom acoustic model](#)
- [Using custom acoustic and custom language models together](#)

Training failures

Training can fail to start for the following reasons:

- The service is currently handling another request for the custom model, such as another training request or a request to add audio resources to the model.
- The custom model contains less than 10 minutes of audio that includes speech, not silence.
- The custom model contains more than 50 hours of audio (for IBM Cloud) or more than 200 hours of audio (for IBM Cloud Pak for Data). **Note:** For IBM Cloud, the maximum hours of audio for a custom acoustic model was reduced from 200 to 50 hours in August and September 2022. For more information, see [Maximum hours of audio](#) .
- You passed a custom language model with the `custom_language_model_id` query parameter that is not in the available state. A custom language model must be fully trained and available to be used to train a custom acoustic

model.

- You passed an incompatible custom language model with the `custom_language_model_id` query parameter. Both custom models must be based on the same version of the same base model.
- The custom model contains one or more invalid audio resources. You can correct the invalid audio resources or set the `strict` parameter to `false` to exclude the invalid resources from the training. The model must contain at least one valid resource for training to succeed.

Request

parameters

customization_id

Required *

str

The customization ID (GUID) of the custom acoustic model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

Example request for IBM Cloud



custom_language_model_id

str

The customization ID (GUID) of a custom language model that is to be used during training of the custom acoustic model. Specify a custom language model that has been trained with verbatim transcriptions of the audio resources or that contains words that are relevant to the contents of the audio resources. The custom language model must be based on the same version of the same base model as the custom acoustic model, and the custom language model must be fully trained and available. The credentials specified with the request must own both custom models.

```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

speech_to_text.train_acous
tic_model('{customization_
id}')
# Poll for acoustic model
status.
```



Example request for IBM Cloud Pak for Data 

strict

bool

If `false` , allows training of the custom acoustic model to proceed as long as the model contains at least one valid audio resource. The method returns an array of `TrainingWarning` objects that lists any invalid resources. By default (`true`), training of a custom acoustic model fails (status code 400) if the model contains one or more invalid audio resources.

Default: `true`

Response

TrainingResponseThe response from training of a custom language or custom acoustic model.

Example responses

Success example

`{}`

warnings

An array of `TrainingWarning` objects that lists any invalid resources contained in the custom model. For custom language models, invalid resources are grouped and identified by type of resource. The method can return warnings only if the `strict` parameter is set to `false`.

> [List\[TrainingWarning\]](#)

Status Code

200 OK. Training of the custom acoustic model started successfully.

400 **Bad Request.** A required parameter is null or invalid, or the custom model is not ready to be trained. Specific failure messages include:

- `No input data modified since last training`
- `The following audio resources are invalid: '{resources}'. Fix errors`

before training.

- Malformed GUID:
'{customization_id}'
 - The specified custom language model
'{customization_id}' is not ready for AM training and/or upgrade. Please make sure it is trained and available.
 - Failed to train. No base model version found in the catalog to match amVersion='{base_model_version}' of the acoustic custom model
'{customization_id}' and lmVersion='{base_model_version}' of passed language custom model
'{customization_id}'.
Upgrading the acoustic custom model may help.
-

401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none">Invalid customization_id '{customization_id}' for user
409	Conflict. The service is currently busy handling a previous request for the custom model: <ul style="list-style-type: none">Customization '{customization_id}' is currently locked to process your last request.
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Reset a custom acoustic model

Resets a custom acoustic model by removing all audio resources from the model. Resetting a custom acoustic model initializes the model to its state when it was first created. Metadata such as the name and language of the model are preserved, but the model's audio resources are removed and must be re-created. The service cannot reset a model while it is handling another request for the model. The service cannot accept subsequent requests for the model until the existing reset request completes. You must use credentials for the instance of the service that owns a model to reset it.

Note: Acoustic model customization is supported only for use with previous-generation models. It is not supported for next-generation models.

See also: [Resetting a custom acoustic model](#)

Request

parameters

```
reset_acoustic_model(  
    self,  
    customization_id:  
    str,  
    **kwargs,  
) -> DetailedResponse
```



Example request for IBM Cloud




customization_id

Required *

str

The customization ID (GUID) of the custom acoustic model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.



```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

speech_to_text.reset_acous
tic_model('{customization_
id}')
```

Example request for IBM Cloud Pak for Data 

Response

Status Code

Example responses

200 **OK.** The custom acoustic model was successfully reset.

400 **Bad Request.** The specified customization ID is invalid:

- **Malformed GUID:**
`'{customization_id}'`

401 **Unauthorized.** The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials:

- **Invalid customization_id**
`'{customization_id}'` for user

409 **Conflict.** The service is currently busy handling a previous request for the custom model:

- **Customization**
`'{customization_id}'` is currently locked to process your last request.

500 **Internal Server Error.** The service experienced an internal error.

503 **Service Unavailable.** The service is currently unavailable.

Success example

`{ }`

Upgrade a custom acoustic model

Initiates the upgrade of a custom acoustic model to the latest version of its base language model. The upgrade method is asynchronous. It can take on the order of minutes or hours to complete depending on the amount of data in the custom model and the current load on the service; typically, upgrade takes approximately twice the length of the total audio contained in the custom model. A custom model must be in the `ready` or `available` state to be upgraded. You must use credentials for the instance of the service that owns a model to upgrade it.

The method returns an HTTP 200 response code to indicate that the upgrade process has begun successfully. You can monitor the status of the upgrade by using the [Get a custom acoustic model](#) method to poll the model's status. The method returns an `AcousticModel` object that includes `status` and `progress` fields. Use a loop to check the status once a minute.

While it is being upgraded, the custom model has the status `upgrading`. When the upgrade

```
upgrade_acoustic_model(  
    self,  
    customization_id:  
    str,  
    *,  
  
    custom_language_model_id:  
    str = None,  
    force: bool =  
    None,  
    **kwargs,  
) -> DetailedResponse
```

is complete, the model resumes the status that it had prior to upgrade. The service cannot upgrade a model while it is handling another request for the model. The service cannot accept subsequent requests for the model until the existing upgrade request completes.

If the custom acoustic model was trained with a separately created custom language model, you must use the `custom_language_model_id` parameter to specify the GUID of that custom language model. The custom language model must be upgraded before the custom acoustic model can be upgraded. Omit the parameter if the custom acoustic model was not trained with a custom language model.

Note: Acoustic model customization is supported only for use with previous-generation models. It is not supported for next-generation models.

See also: [Upgrading a custom acoustic model](#)

Request

parameters

Example request for IBM Cloud



customization_id

Required *

str

The customization ID (GUID) of the custom acoustic model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

custom_language_model_id

str

If the custom acoustic model was trained with a custom language model, the customization ID (GUID) of that custom language model. The custom language model must be upgraded before the custom acoustic model can be upgraded. The custom language model must be fully trained and available. The credentials specified with the request must own both custom models.

```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator
```

```
authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(
```

```
authenticator=authenticato
r
)
```

```
speech_to_text.set_service
_url('{url}')
```

```
speech_to_text.upgrade_aco
ustic_model('{customizatio
n_id}')
# Poll for acoustic model
status.
```

Example request for IBM Cloud Pak for Data 

force

bool

If `true` , forces the upgrade of a custom acoustic model for which no input data has been modified since it was last trained. Use this parameter only to force the upgrade of a custom acoustic model that is trained with a custom language model, and only if you receive a 400 response code and the message `No input data modified since last training` . See [Upgrading a custom acoustic model](#) .

Default: `false`

Response

Status Code

200	OK. Upgrade of the custom acoustic model started successfully.
400	Bad Request. A parameter is null or invalid, or the specified custom model cannot be upgraded. Specific failure messages include: <ul style="list-style-type: none"><code>Malformed GUID:</code>

Example responses

Success example

`{}`

'{customization_id}'

- Custom model is up-to-date
- No input data available to upgrade the model
- No input data modified since last training
- Cannot upgrade failed custom model
- The passed language custom model needs to be upgraded in order to upgrade the acoustic custom model.
- The specified custom language model '{customization_id}' is not ready for AM training and/or upgrade. Please make sure it is trained and available.
- Base model name mismatch detected. Please make sure that the base model name of the language custom model matches the base model name of the acoustic custom model.
- Invalid model type for

```
customization_id  
'{customization_id}'
```

401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none">• <code>Invalid customization_id '{customization_id}' for user</code>
------------	---

409	Conflict. The service is currently busy handling a previous request for the custom model: <ul style="list-style-type: none">• <code>Customization '{customization_id}' is currently locked to process your last request.</code>
------------	--

500	Internal Server Error. The service experienced an internal error.
------------	--

503	Service Unavailable. The service is currently unavailable.
------------	---

List audio resources

Lists information about all audio resources from a custom acoustic model. The information includes the name of the resource and information about its audio data, such as its duration. It also includes the status of the audio resource, which is important for checking the service's analysis of the resource in response to a request to add it to the custom acoustic model. You must use credentials for the instance of the service that owns a model to list its audio resources.

Note: Acoustic model customization is supported only for use with previous-generation models. It is not supported for next-generation models.

See also: [Listing audio resources for a custom acoustic model](#) .

Request

parameters

```
list_audio(  
    self,  
    customization_id:  
str,  
    **kwargs,  
) -> DetailedResponse
```

Example request for IBM Cloud

customization_id

Required *

str

The customization ID (GUID) of the custom acoustic model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

audio_resources =
speech_to_text.list_audio(
'{customization_id}').get_
result()
print(json.dumps(audio_res
ources, indent=2))
```

Example request for IBM Cloud Pak for Data 

Response

AudioResources Information about the audio resources from a custom acoustic model.

total_minutes_of_audio

Always included *

float

The total minutes of accumulated audio summed over all of the valid audio resources for the custom acoustic model. You can use this value to determine whether the custom model has too little or too much audio to begin training.

audio

Always included *

An array of **AudioResource** objects that provides information about the audio resources of the custom acoustic model. The array is empty if the custom model has no audio resources.

> [List\[AudioResource\]](#)

Example responses

Success example

```
{  
  
  "total_minutes_of_audio":  
    11.45,  
  "audio": [  
    {  
      "duration": 131,  
      "name": "audio1",  
      "details": {  
        "codec":  
          "pcm_s16le",  
        "type": "audio",  
        "frequency": 22050  
      },  
      "status": "ok"  
    },  
    {  
      "duration": 556,  
      "name": "audio2",  
      "details": {  
        "type": "archive",  
        "compression":  
          "zip"  
      },  
      "status": "ok"  
    },  
  ],  
}
```

Status Code

200	OK. The request succeeded.
400	Bad Request. The specified customization ID is invalid: <ul style="list-style-type: none">Malformed GUID: '{customization_id}'
401	Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials: <ul style="list-style-type: none">Invalid customization_id '{customization_id}' for user
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

Add an audio resource

Adds an audio resource to a custom acoustic model. Add audio content that reflects the

```
{
  "duration": 0,
  "name": "audio3",
  "details": {},
  "status":
    "being_processed"
}
```

```
add_audio(
    self,
```



acoustic characteristics of the audio that you plan to transcribe. You must use credentials for the instance of the service that owns a model to add an audio resource to it. Adding audio data does not affect the custom acoustic model until you train the model for the new data by using the [Train a custom acoustic model](#) method.

You can add individual audio files or an archive file that contains multiple audio files. Adding multiple audio files via a single archive file is significantly more efficient than adding each file individually. You can add audio resources in any format that the service supports for speech recognition.

You can use this method to add any number of audio resources to a custom model by calling the method once for each audio or archive file. You can add multiple different audio resources at the same time. You must add a minimum of 10 minutes of audio that includes speech, not just silence, to a custom acoustic model before you can train it. No audio resource, audio- or archive-type, can be larger than 100 MB. To add an audio resource that has the same name as an existing audio resource, set the `allow_overwrite` parameter to `true`; otherwise, the request fails. A custom model can contain no more than 50 hours of audio (for IBM Cloud) or 200 hours of audio (for IBM Cloud Pak for Data). **Note:** For IBM Cloud, the

```
customization_id:
str,
audio_name: str,
audio_resource:
BinaryIO,
*,
content_type: str
= None,

contained_content_type:
str = None,
allow_overwrite:
bool = None,
**kwargs,
) -> DetailedResponse
```


maximum hours of audio for a custom acoustic model was reduced from 200 to 50 hours in August and September 2022. For more information, see [Maximum hours of audio](#) .

The method is asynchronous. It can take several seconds or minutes to complete depending on the duration of the audio and, in the case of an archive file, the total number of audio files being processed. The service returns a 201 response code if the audio is valid. It then asynchronously analyzes the contents of the audio file or files and automatically extracts information about the audio such as its length, sampling rate, and encoding. You cannot submit requests to train or upgrade the model until the service's analysis of all audio resources for current requests completes.

To determine the status of the service's analysis of the audio, use the [Get an audio resource](#) method to poll the status of the audio. The method accepts the customization ID of the custom model and the name of the audio resource, and it returns the status of the resource. Use a loop to check the status of the audio every few seconds until it becomes `ok` .

Note: Acoustic model customization is supported only for use with previous-generation models. It is not supported for next-generation models.

See also: [Add audio to the custom acoustic model](#) .

Content types for audio-type resources

You can add an individual audio file in any format that the service supports for speech recognition. For an audio-type resource, use the `Content-Type` parameter to specify the audio format (MIME type) of the audio file, including specifying the sampling rate, channels, and endianness where indicated.

- `audio/alaw` (Specify the sampling rate (`rate`) of the audio.)
- `audio/basic` (Use only with narrowband models.)
- `audio/flac`
- `audio/g729` (Use only with narrowband models.)
- `audio/l16` (Specify the sampling rate (`rate`) and optionally the number of channels (`channels`) and endianness (`endianness`) of the audio.)
- `audio/mp3`
- `audio/mpeg`
- `audio/mulaw` (Specify the sampling rate (`rate`) of the audio.)

- `audio/ogg` (The service automatically detects the codec of the input audio.)
- `audio/ogg;codecs=opus`
- `audio/ogg;codecs=vorbis`
- `audio/wav` (Provide audio with a maximum of nine channels.)
- `audio/webm` (The service automatically detects the codec of the input audio.)
- `audio/webm;codecs=opus`
- `audio/webm;codecs=vorbis`

The sampling rate of an audio file must match the sampling rate of the base model for the custom model: for broadband models, at least 16 kHz; for narrowband models, at least 8 kHz. If the sampling rate of the audio is higher than the minimum required rate, the service down-samples the audio to the appropriate rate. If the sampling rate of the audio is lower than the minimum required rate, the service labels the audio file as `invalid`.

See also: [Supported audio formats](#) .

Content types for archive-type resources

You can add an archive file (`.zip` or `.tar.gz` file) that contains audio files in any format that the service supports for speech recognition. For an

archive-type resource, use the `Content-Type` parameter to specify the media type of the archive file:

- `application/zip` for a **.zip** file
- `application/gzip` for a **.tar.gz** file.

When you add an archive-type resource, the `Contained-Content-Type` header is optional depending on the format of the files that you are adding:

- For audio files of type `audio/alaw` , `audio/basic` , `audio/l16` , or `audio/mulaw` , you must use the `Contained-Content-Type` header to specify the format of the contained audio files. Include the `rate` , `channels` , and `endianness` parameters where necessary. In this case, all audio files contained in the archive file must have the same audio format.
- For audio files of all other types, you can omit the `Contained-Content-Type` header. In this case, the audio files contained in the archive file can have any of the formats not listed in the previous bullet. The audio files do not need to have the same format.

Do not use the `Contained-Content-Type` header when adding an audio-type resource.

Naming restrictions for embedded audio files

The name of an audio file that is contained in an archive-type resource can include a maximum of 128 characters. This includes the file extension and all elements of the name (for example, slashes).

Request

parameters	
customization_id Required * str The customization ID (GUID) of the custom acoustic model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.	
audio_name Required * str	The name of the new audio resource for the custom acoustic model. Use a localized name that matches the language of the custom model and reflects the contents of the resource.

Example request for IBM Cloud

```
from os.path import join,
dirname
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
```

- Include a maximum of 128 characters in the name.
- Do not use characters that need to be URL-encoded. For example, do not use spaces, slashes, backslashes, colons, ampersands, double quotes, plus signs, equals signs, questions marks, and so on in the name. (The service does not prevent the use of these characters. But because they must be URL-encoded wherever used, their use is strongly discouraged.)
- Do not use the name of an audio resource that has already been added to the custom model.

```
_url('{url}')
```

```
with  
open(join(dirname(__file__  
) , './.', 'audio1.wav'),  
      'rb') as  
audio_file:  
  
speech_to_text.add_audio(  
  
    '{customization_id}',  
    'audio1',  
    audio_file,  
  
    content_type='audio/wav'  
    )  
# Poll for audio status.
```

Example request for IBM Cloud Pak for Data 

audio_resource The audio resource that is to
Required ^{*} be added to the custom
BinaryIO acoustic model, an individual
audio file or an archive file.

With the `curl` command,
use the `--data-binary`
option to upload the file for
the request.

content_type For an audio-type resource,
str the format (MIME type) of
the audio. For more
information, see **Content
types for audio-type
resources** in the method
description.

For an archive-type resource,
the media type of the archive
file. For more information,
see **Content types for
archive-type resources** in
the method description.

Allowable values: [
`application/zip` ,
`application/gzip` ,
`audio/alaw` ,
`audio/basic` ,
`audio/flac` ,
`audio/g729` , `audio/l16` ,

```
audio/mp3 , audio/mpeg ,
audio/mulaw ,
audio/ogg ,
audio/ogg;codecs=opus
,
audio/ogg;codecs=vorbis
, audio/wav , audio/webm
,
audio/webm;codecs=opus
,
audio/webm;codecs=vorbis
]
```

contained_content_type

str

For an archive-type resource,

specify the format of the audio files that are contained in the archive file if they are of type `audio/alaw` , `audio/basic` , `audio/l16` , or `audio/mulaw` . Include the `rate` , `channels` , and `endianness` parameters where necessary. In this case, all audio files that are contained in the archive file must be of the indicated

type.

For all other audio formats, you can omit the header. In this case, the audio files can be of multiple types as long as they are not of the types listed in the previous paragraph.

The parameter accepts all of the audio formats that are supported for use with speech recognition. For more information, see **Content types for audio-type resources** in the method description.

For an audio-type resource, omit the header.

Allowable values: [

`audio/alaw` ,
`audio/basic` ,
`audio/flac` ,
`audio/g729` , `audio/l16` ,
`audio/mp3` , `audio/mpeg` ,
`audio/mulaw` ,
`audio/ogg` ,
`audio/ogg;codecs=opus`
,

```
audio/ogg;codecs=vorbis
, audio/wav , audio/webm
,
audio/webm;codecs=opus
,
audio/webm;codecs=vorbis
]
```

allow_overwrite If `true` , the specified audio resource overwrites an existing audio resource with the same name. If `false` , the request fails if an audio resource with the same name already exists. The parameter has no effect if an audio resource with the same name does not already exist.

Default: `false`

Response

Status Code

201 **Created.** Addition of the audio resource was successfully started. The service is analyzing the data.

Example responses

Success example



400

Bad Request. A required parameter is null or invalid, the specified customization ID or audio resource is invalid, or the specified audio resource already exists. Specific failure messages include:

- Malformed GUID:
'{customization_id}'
- Audio file not specified or empty
- Invalid audio format detected
- Invalid or missing audio content type
- Audio '{name}' already exists - change its name, remove existing file before adding new one, or overwrite existing file by setting 'allow_overwrite' flag to 'true'

{ }



401	<p>Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials:</p> <ul style="list-style-type: none">• <code>Invalid customization_id '{customization_id}' for user</code>
405	<p>Method Not Allowed. The audio resource name includes characters that need to be URL-encoded.</p>
409	<p>Conflict. The service is currently busy handling a previous request for the custom model:</p> <ul style="list-style-type: none">• <code>Customization '{customization_id}' is currently locked to process your last request.</code>
500	<p>Internal Server Error. An internal error prevented the service from satisfying the request. You can also receive status code 500 <code>Forwarding Error</code> if the service is currently busy handling a previous request for the custom model.</p>
503	<p>Service Unavailable. The service is currently unavailable.</p>

Get an audio resource

Gets information about an audio resource from a custom acoustic model. The method returns an `AudioListing` object whose fields depend on the type of audio resource that you specify with the method's `audio_name` parameter:

- *For an audio-type resource*, the object's fields match those of an `AudioResource` object: `duration`, `name`, `details`, and `status`.
- *For an archive-type resource*, the object includes a `container` field whose fields match those of an `AudioResource` object. It also includes an `audio` field, which contains an array of `AudioResource` objects that provides information about the audio files that are contained in the archive.

The information includes the status of the specified audio resource. The status is important for checking the service's analysis of a resource that you add to the custom model.

- *For an audio-type resource*, the `status` field is located in the `AudioListing` object.
- *For an archive-type resource*, the `status` field is located in the `AudioResource` object that is returned in the `container` field.

```
get_audio(  
    self,  
    customization_id:  
    str,  
    audio_name: str,  
    **kwargs,  
) -> DetailedResponse
```



You must use credentials for the instance of the service that owns a model to list its audio resources.

Note: Acoustic model customization is supported only for use with previous-generation models. It is not supported for next-generation models.

See also: [Listing audio resources for a custom acoustic model](#) .

Request

parameters
<div><div>customization_id</div><div>Required *</div><div>str</div><div>The customization ID (GUID) of the custom acoustic model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.</div></div>

Example request for IBM Cloud



audio_name

Required *

str

The name of the audio resource for the custom acoustic model.

```
import json
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

audio_listing =
speech_to_text.get_audio(
    '{customization_id}',
    'audio2'
).get_result()
print(json.dumps(audio_lis
ting, indent=2))
```

Example request for IBM Cloud Pak for Data 

Response

AudioListing	Information about an audio resource from a custom acoustic model.
duration int	For an audio-type resource, the total seconds of audio in the resource. Omitted for an archive-type resource.
name str	For an audio-type resource, the user-specified name of the resource. Omitted for an archive-type resource.
details	For an audio-type resource, an AudioDetails object that provides detailed information about the resource. The object is empty until the service finishes processing the audio. Omitted for an archive-type resource.
> AudioDetails	
status str	For an audio-type resource,

Example responses

Success example

```
{
  "container": {
    "duration": 556,
    "name": "audio2",
    "details": {
      "type": "archive",
      "compression": "zip"
    },
    "status": "ok"
  },
  "audio": [
    {
      "duration": 121,
      "name": "audio-
file1.wav",
      "details": {
        "codec":
"pcm_s16le",
        "type": "audio",
        "frequency": 16000
      },
      "status": "ok"
    },
    {
      "duration": 133,
```


the status of the resource:

- **ok** : The service successfully analyzed the audio data. The data can be used to train the custom model.
- **being_processed** : The service is still analyzing the audio data. The service cannot accept requests to add new audio resources or to train the custom model until its analysis is complete.
- **invalid** : The audio data is not valid for training the custom model (possibly because it has the wrong format or sampling rate, or because it is corrupted).

Omitted for an archive-type resource.

Possible values: [`ok` ,
`being_processed` ,
`invalid`]

```

    "name": "audio-
file2.wav",
    "details": {
        "codec":
"pcm_s16le",
        "type": "audio",
        "frequency": 16000
    },
    "status": "ok"
},
{
    "duration": 112,
    "name": "audio-
file3.wav",
    "details": {
        "codec":
"pcm_s16le",
        "type": "audio",
        "frequency": 16000
    },
    "status": "ok"
},
{
    "duration": 129,
    "name": "audio-
file4.wav",
    "details": {
        "codec":
"pcm_s16le",
        "type": "audio",
        "frequency": 16000
    },
    "status": "ok"
},

```

container *For an archive-type resource, an object of type `AudioResource` that provides information about the resource. Omitted for an audio-type resource.*

> [AudioResource](#)

audio *For an archive-type resource, an array of `AudioResource` objects that provides information about the audio-type resources that are contained in the resource. Omitted for an audio-type resource.*

> [List\[AudioResource\]](#)

Status Code

200 OK. The request succeeded.

```
{
  "duration": 61,
  "name": "audio-
file5.wav",
  "details": {
    "codec":
"pcm_s16le",
    "type": "audio",
    "frequency": 16000
  },
  "status": "ok"
}
]
```

400 **Bad Request.** The specified customization ID or audio resource name is invalid, including the case where the audio resource does not exist for the custom model. Specific failure messages include:

- `Malformed GUID: '{customization_id}'`
- `Invalid value for audio name '{audio_name}'`

401 **Unauthorized.** The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials:

- `Invalid customization_id '{customization_id}' for user`

500 **Internal Server Error.** The service experienced an internal error.

503 **Service Unavailable.** The service is currently unavailable.

Delete an audio resource

Deletes an existing audio resource from a custom acoustic model. Deleting an archive-type audio resource removes the entire archive of files. The service does not allow deletion of individual files from an archive resource.

Removing an audio resource does not affect the custom model until you train the model on its updated data by using the [Train a custom acoustic model](#) method. You can delete an existing audio resource from a model while a different resource is being added to the model. You must use credentials for the instance of the service that owns a model to delete its audio resources.

Note: Acoustic model customization is supported only for use with previous-generation models. It is not supported for next-generation models.

See also: [Deleting an audio resource from a custom acoustic model](#) .

Request

parameters

```
delete_audio(  
    self,  
    customization_id:  
    str,  
    audio_name: str,  
    **kwargs,  
) -> DetailedResponse
```



customization_id

Required *

str

The customization ID (GUID) of the custom acoustic model that is to be used for the request. You must make the request with credentials for the instance of the service that owns the custom model.

audio_name

Required *

str

The name of the audio resource for the custom acoustic model.

```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator
```

```
authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(
```

```
authenticator=authenticato
r
)
```

```
speech_to_text.set_service
_url('{url}')
```

```
speech_to_text.delete_audi
o(
    '{customization_id}',
    'audio1'
)
```

Example request for IBM Cloud Pak for Data

Response

Status Code

Status code

200

OK. The audio resource was successfully deleted from the custom acoustic model.

400

Bad Request. The specified customization ID or audio resource name is invalid, including the case where the audio resource does not exist for the custom model. Specific failure messages include:

- **Malformed GUID:**
`'{customization_id}'`
- **Invalid value for audio name** `'{audio_name}'`

401

Unauthorized. The specified credentials are invalid or the specified customization ID is invalid for the requesting credentials:

- **Invalid customization_id** `'{customization_id}'` for user

405

Method Not Allowed. No audio resource name was specified with the request.

Example responses

Success example

```
{}
```

409

Conflict. The service is currently busy handling a previous request for the custom model:

- Customization
'{customization_id}' is currently locked to process your last request.

500

Internal Server Error. The service experienced an internal error.

503

Service Unavailable. The service is currently unavailable.

Delete labeled data

Deletes all data that is associated with a specified customer ID. The method deletes all data for the customer ID, regardless of the method by which the information was added. The method has no effect if no data is associated with the customer ID. You must issue the request with credentials for the same instance of the service that was used to associate the customer ID with the data. You associate a customer ID with data by passing the `X-Watson-Metadata` header with a request that passes the data.

Note: If you delete an instance of the service from the service console, all data associated with that service instance is automatically deleted. This includes all custom language models, corpora, grammars, and words; all custom acoustic models and audio resources; all registered endpoints for the asynchronous HTTP interface; and all data related to speech recognition requests.

See also: [Information security](#) .

Request

parameters

```
delete_user_data(  
    self,  
    customer_id: str,  
    **kwargs,  
) -> DetailedResponse
```



customer_id

Required ^{*}
str

The customer ID for which
all data is to be deleted.

```
from ibm_watson import
SpeechToTextV1
from
ibm_cloud_sdk_core.authent
icators import
IAMAuthenticator

authenticator =
IAMAuthenticator('{apikey}
')
speech_to_text =
SpeechToTextV1(

authenticator=authenticato
r
)

speech_to_text.set_service
_url('{url}')

speech_to_text.delete_user
_data('{customer_id}')
```

Example request for IBM Cloud Pak for Data 

Response

Status Code

200	OK. The deletion request was successfully submitted.
400	Bad Request. The request did not pass a customer ID: <ul style="list-style-type: none">• No customer ID found in the request
500	Internal Server Error. The service experienced an internal error.
503	Service Unavailable. The service is currently unavailable.

[Privacy statement](#)

<