Automatic Speech Recognition

API Documentation

Product Documentation

Contents

API Documentation
Real-Time Speech Recognition APIs
Real-Time Speech Recognition (WebSocket)

Last updated：2024-04-16 15:22:19

Note:
This API is on v2.0 and is different from API 3.0 in terms of parameter styles and error codes.

API Description

This API is used to recognize a real-time audio stream over the WebSocket protocol and return the recognition result synchronously. It achieves the effect of instant speech-to-text conversion.
Before using this API, you need to activate the service in the ASR console first. Then, go to the Manage API Key page to create a key and generate `AppID` , `SecretID` , and `SecretKey` required for calculating authentication signatures during API call.

API Requirements

When integrating the real-time speech recognition API, you need to comply with the following requirements:

Item	Description
Language	Supported languages include Mandarin, English, Korean, Japanese, Thai, Bahasa, Vietnamese, Malay, Filipino, Portuguese, Turkish, Arabic, Spanish, Hindi, French and German. You can set the language through the `engine_model_type` parameter.
Industry	General, finance, gaming, education, and healthcare.
Audio attribute	Sample rate: 16000 Hz or 8000 Hz Bit depth: 16 bits Channel: mono
Audio format	PCM, WAV, Opus, Speex, SILK, MP3, M4A, and AAC
Request protocol	WSS
Request address	wss://asr.cloud.tencent.com/asr/v2/<appid>?{request parameter}
API authentication	Signature-based authentication. For more information, see Signature generation.

Response format	JSON
Data sending	We recommend you send a packet of 40 ms in length every 40 ms (i.e., real-time factor (RTF) of 1:1). The corresponding PCM is 640 bytes at a sample rate of 8 kHz or 1,280 bytes at a sample rate of 16 kHz. An audio sending rate exceeding the 1:1 RTF or an audio packet sending interval exceeding 6 seconds may cause an error in the engine. In this case, the backend will return an error and close the connection.
Concurrency limit	The number of concurrent connections per account is 50. If you need more concurrent connections, submit a ticket for application.

API Call Process

The API call process can be divided into two phases: handshake and recognition. In both phases, the backend returns a text message containing a serialized JSON string in the following format:

Field	Type	Description
code	Integer	The status code. 0: normal; other values: an error occurred.
message	String	The error message. This field explains the cause of the error. Note that the returned messages are subject to changes as the service is updated or optimized.
voice_id	String	The unique ID of the audio stream, which is generated by the client during the handshake and assigned in the call parameters.
message_id	String	The unique ID of the message.
result	Result	The latest speech recognition result.
final	Integer	If this field returns `1` , all of the audio stream has been recognized.

The structure of the recognition result `Result` is in the following format:

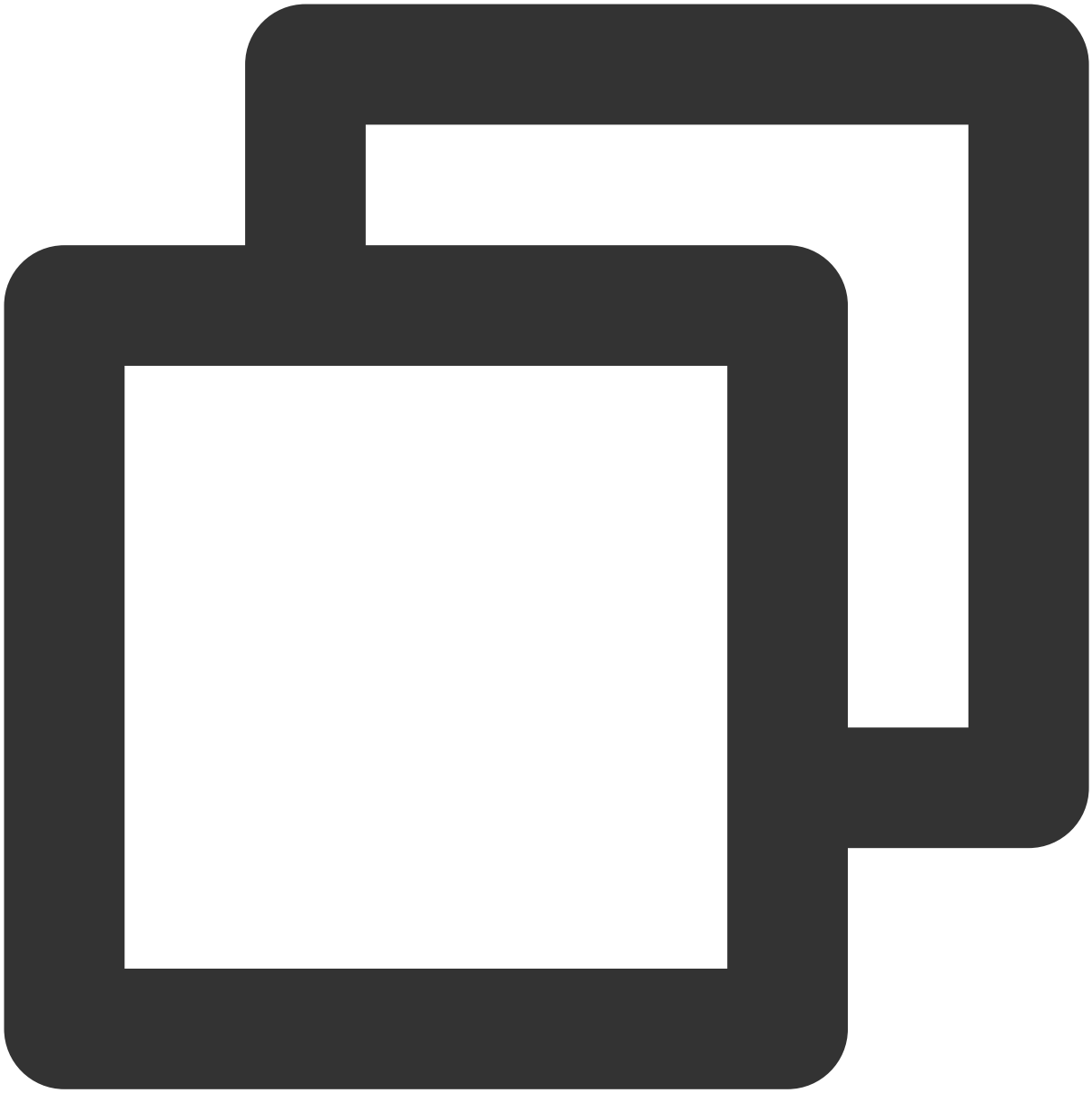
Field	Type	Description
slice_type	Integer	The recognition result type. 0: started recognizing a paragraph 1: recognizing a paragraph, where `voice_text_str` indicates an unstable result (the recognition result of the paragraph may change) 2: stopped recognizing a paragraph, where `voice_text_str` indicates a stable result (the recognition result of the paragraph will not

		change) Depending on the sent audio, the following `slice_type` sequence may be returned: 0-1-2: started recognizing a paragraph, recognizing a paragraph (`1` may be returned multiple times), stopped recognizing a paragraph 0-2: started recognizing a paragraph, stopped recognizing a paragraph 2: returned the complete recognition result of a paragraph
index	Integer	The number of the result of the current paragraph in the entire audio stream, which increases from 0.
start_time	Integer	The start time of the result of the current paragraph in the entire audio stream
end_time	Integer	The end time of the result of the current paragraph in the entire audio stream
voice_text_str	String	The text result of the current paragraph encoded in UTF8
word_size	Integer	The number of word results of the current paragraph
word_list	Word Array	The list of words contained in the current paragraph. The structure of a word is in the following format: word: the content of the word in string type start_time: the start time of the word in the entire audio stream in integer type end_time: the end time of the word in the entire audio stream in integer type stable_flag: the status of the result in integer type. 0: unstable; 1: stable.

Handshake phase

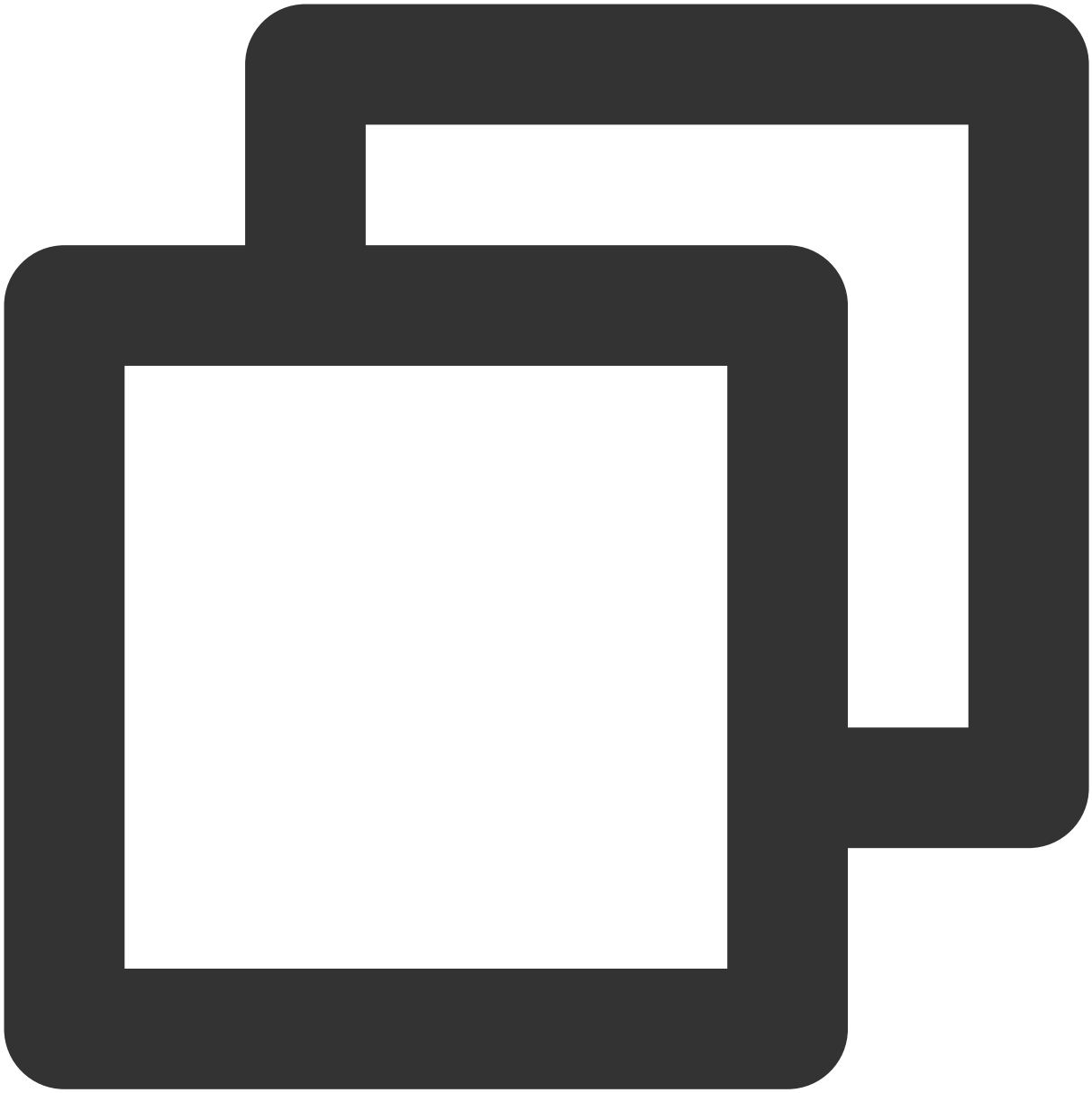
Request format

During the handshake, the client initiates a WebSocket connection request with a URL in the following format:



```
wss://asr.cloud.tencent.com/asr/v2/<appid>?{request parameter}
```

Here, the <appid> needs to be replaced with the `AppID` of your Tencent Cloud account, which can be obtained on the Manage API Key page. The format of the {request parameter} is as follows:



```
key1=value2&key2=value2... (both `key` and `value` need to be URL-encoded)
```

Parameter description:

Parameter	Required	Type	Description
secretid	Yes	String	The `SecretId` of the key of your Tencent Cloud account, which can be obtained on the Manage API Key page.
timestamp	Yes	Integer	The current UNIX timestamp in seconds. If the

			difference between the UNIX timestamp and the current time is too large, a signature expiration error may occur.
expired	Yes	Integer	The UNIX timestamp of the signature expiration time in seconds. `expired` must be greater than `timestamp`, and `expired` - `timestamp` must be smaller than 90 days.
nonce	Yes	Integer	A random positive integer, which you need to generate on your own and can contain up to ten digits.
engine_model_type	Yes	String	The engine model type For phone call scenarios: • 8k_zh：8kHz, for Mandarin in general scenarios, • 8k_en：8kHz, for English; For non-phone call scenarios: • 16k_zh:16 kHz, for Mandarin in general scenarios; • 16k_en:16 kHz, for English; • 16k_ko:16 kHz, for Korean; • 16k_ja:16 kHz, for Japanese; • 16k_th:16 kHz, for Thai; • 16k_id:16 kHz, for Bahasa; • 16k_vi:16kHz, for Vietnamese; • 16k_ms:16kHz, for Malay; • 16k_fil:16kHz, for Filipino; • 16k_pt:16kHz, for Portuguese; • 16k_tr:16kHz, for Turkish; • 16k_ar:16kHz, for Arabic; • 16k_es:16kHz, for spanish; • 16k_hi:16kHz, for Hindi; • 16k_fr:16kHz, for French; • 16k_de:16kHz, for German;
voice_id	Yes	String	The unique identifier of each audio, which you need to generate on your own.
voice_format	No	Int	The audio encoding format, which is optional. The default value is `4`. 1: PCM; 4: Speex (SP); 6: SILK; 8: MP3; 10: Opus (Opus audio stream encapsulation description); 12: WAV; 14: M4A (each segment must be a complete M4A audio); 16: AAC
needvad	No	Integer	0: disables VAD; 1: enables VAD If an audio segment exceeds 60 seconds in length, you should enable voice activity detection (VAD).
hotword_id	No	String	The ID of the keyword list. If this parameter is set, the

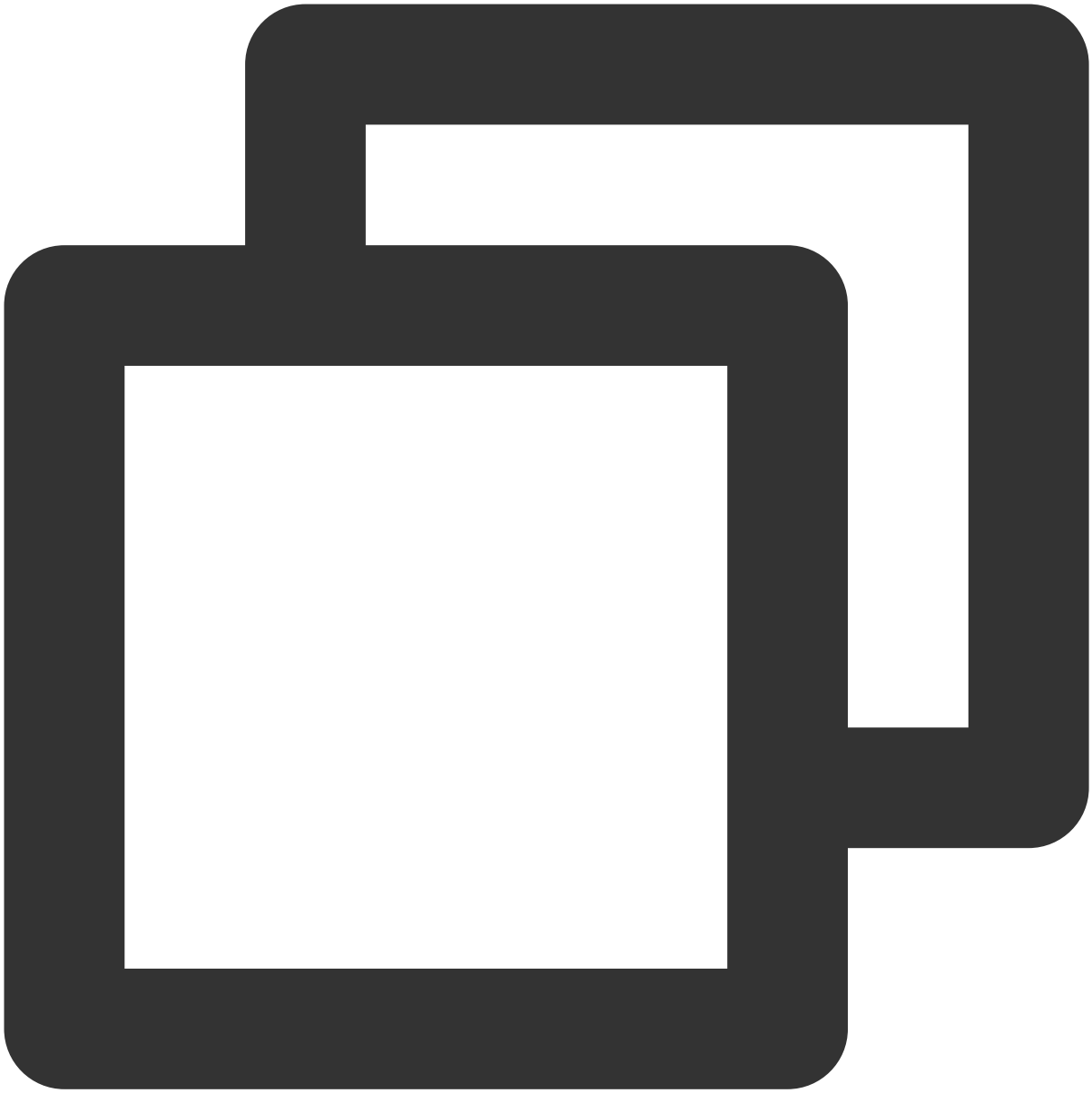
			corresponding list will take effect; otherwise, the default keyword list will be used.
reinforce_hotword	No	Integer	Hot word enhancement function. The default is 0, 0: not enabled, 1: enabled After activation (only 8k_zh, 16k_zh supported), the homophonic replacement function will be enabled, and homophonic words and phrases will be configured in the hot words section.
customization_id	No	String	The ID of the self adaptive learning model. If this parameter is set, the corresponding model will take effect; otherwise, the last launched model will be used.
filter_dirty	No	Integer	Whether to filter restricted words (for the Mandarin engine only). 0 (default value): does not filter; 1: filters; 2: replaces restricted words with "*".
filter_modal	No	Integer	Whether to filter interjections (for the Mandarin engine only). 0 (default value): does not filter; 1: filters; 2: filters strictly.
filter_punc	No	Integer	Whether to filter the period at the end of a sentence (for the Mandarin engine only). 0 (default value): no; 1: yes.
filter_empty_result	NO	Integer	Whether to callback and recognize empty results, default to 1. 0: Callback null result; 1: Do not callback empty results; **Attention**：If slice_type=0 and slice_type=2 need to pair callbacks, filter_empty_result=0 needs to be set. In general, pairing and returning are required in outbound calling scenarios, and slice_type=0 is used to determine whether there is a human voice appearing.
convert_num_mode	No	Integer	Whether to intelligently convert Chinese numbers to Arabic numerals (for the Mandarin engine only). 0: directly outputs Chinese numbers; 1 (default value): intelligently converts based on the scenario; 3: enables mathematic number conversion.
word_info	No	Int	Whether to display the word-level timestamp. 0 (default value): does not display; 1: displays timestamps without punctuation marks; 2: displays timestamps with punctuation marks. Supported engines include 16k_zh, 16k_en, and 16k_ja.
vad_silence_time	No	Integer	The speech segmentation detection threshold in ms. A

			silence longer than this threshold will be considered as segmentation (this parameter is commonly used in customer service scenarios and requires `needvad = 1`). Value range: 240–2000. We recommend you not adjust this parameter; otherwise, the recognition result may be affected. Supported engine is 16k_zh.
max_speak_time	No	Integer	Mandatory sentence breaking function, with a value range of 5000-90000 (in milliseconds) and a default value of 0 (not enabled). In the case of continuous speech without interruption, this parameter will implement forced sentence breaking (the result becomes steady state, slice_type=2). For example, in game commentary scenarios, if the commentator continuously explains without interruption and cannot break a sentence, setting this parameter to 10000 will receive a callback from slice_type=2 every 10 seconds.
noise_threshold	No	Float	The noise parameter threshold is set to 0 by default, with a value range of [-1,1]. For some audio clips, the larger the value, the greater the noise level. The smaller the value, the greater the judgment of human voice. **Caution: may affect recognition effectiveness.**
signature	Yes	String	The API signature parameter.
hotword_list	No	String	Temporary hot word list: This parameter is used to improve recognition accuracy. Single hot word limit: "Hot word weight", with a maximum of 30 characters per hot word and a weight of 1-11, such as "Tencent Cloud |5" or "ASR |11"; Temporary hot word list restriction: Multiple hot words can be separated by English commas, supporting a maximum of 128 hot words, such as "Tencent Cloud | 10, Speech Recognition |5, ASR | 11"; Parameter hotword_id (Hot Word List) and hotword_list (temporary hot word list) difference: hotword_id：Hot word list. You need to first create a hot word list in the console or interface to obtain the corresponding hotword_ Pass in the ID parameter to use the hot word function; hotword_list：Temporary hot word list. Every time a request is made, a temporary hot word list is directly passed in to use the hot word function, and the cloud does not retain the temporary hot word list. Suitable for users with a high demand for hot words;

			Attention： If both hotword_id and hotword_list are passed in at the same time, hotword_list will be used first; When the weight of hot words is set to 11, the current hot words will be upgraded to super hot words. It is recommended to only set important and must be effective hot words to 11. Setting too many hot words with a weight of 11 will affect the overall word accuracy.
input_sample_rate	NO	Interge	Supporting 8k audio in PCM format and upsampling it to 16k for recognition when the engine sampling rate does not match, can effectively improve recognition accuracy.Only supported: 8000. For example, if 8000 is passed in, the PCM audio sampling rate is 8k, and when the engine selects 16k_ Zh, then the PCM audio with a sampling rate of 8k can be at 16k_ Normal recognition under the zh engine. Note: This parameter is only applicable to PCM format audio. If no value is passed in, the default state will be maintained, that is, the engine sampling rate called by default is equal to the PCM audio sampling rate.

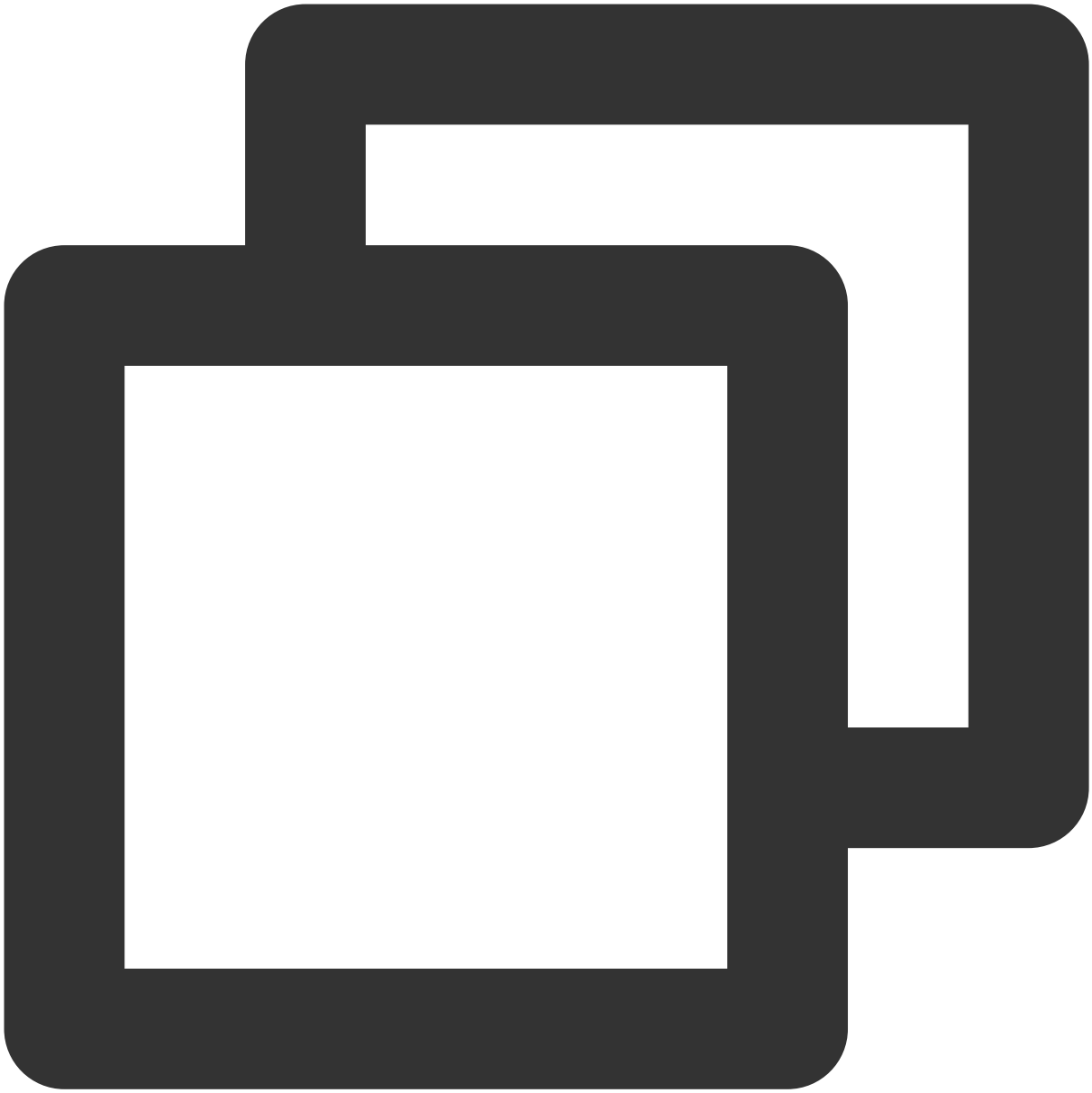
Signature generation

1. Sort all parameters except `signature` in lexicographical order to concatenate them into the request URL as the original signature string. Here, `Appid=1259228442, SecretId=AKIDoQq1zhZMN8dv0psmvud6OUKuGPO7pu0r` is concatenated into the string as follows:



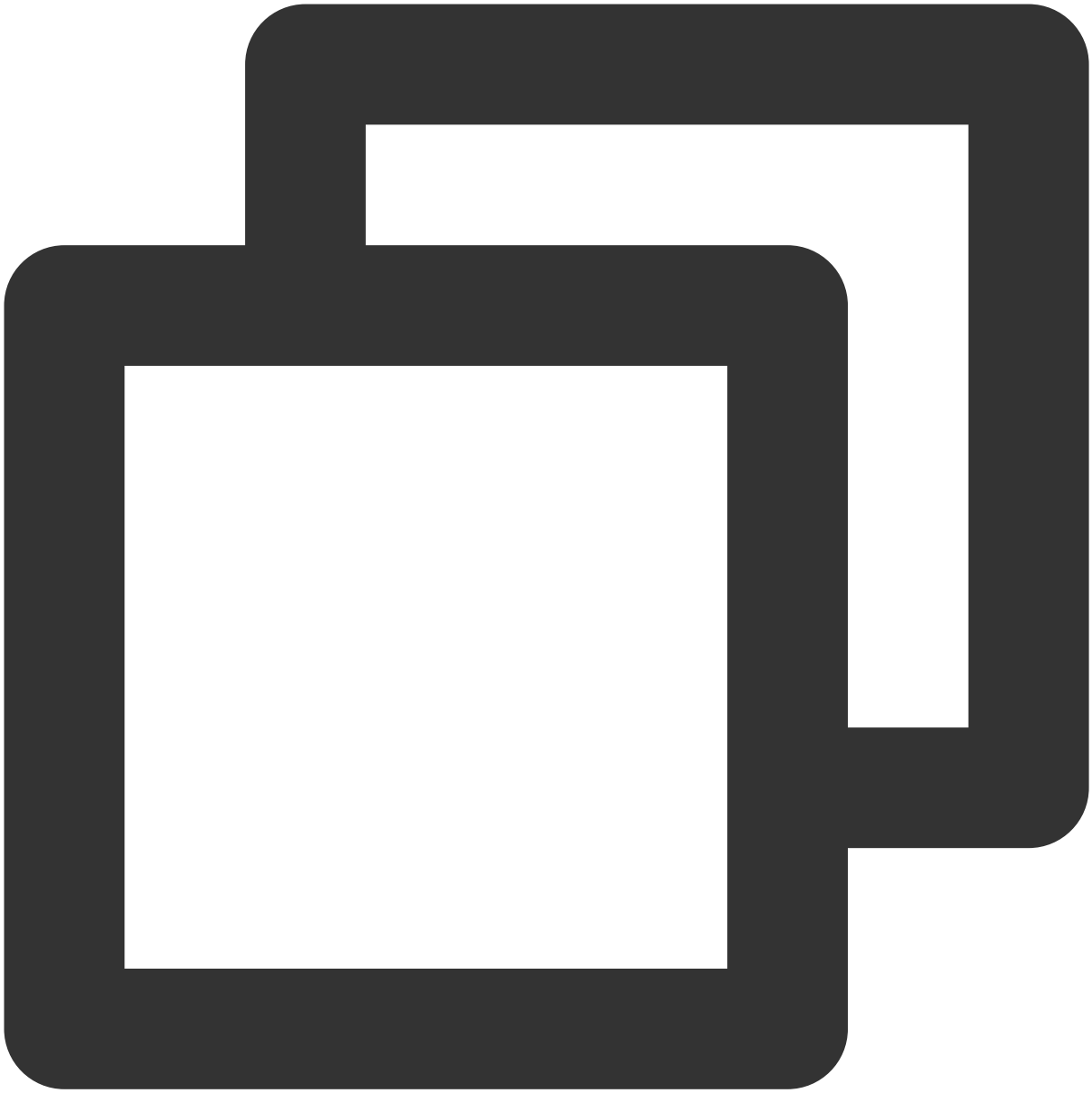
```
asr.cloud.tencent.com/asr/v2/1259228442?engine_model_type=16k_zh&expired=1592380492
```

2. Encrypt the original signature string by using `SecretKey` with HMACSHA1 and then Base64-encode it. For example, perform this operation on the original signature string generated in the previous step with `SecretKey=kFpwoX5RYQ2SkqpeHgqmSzHK7h3A2fni` to get:



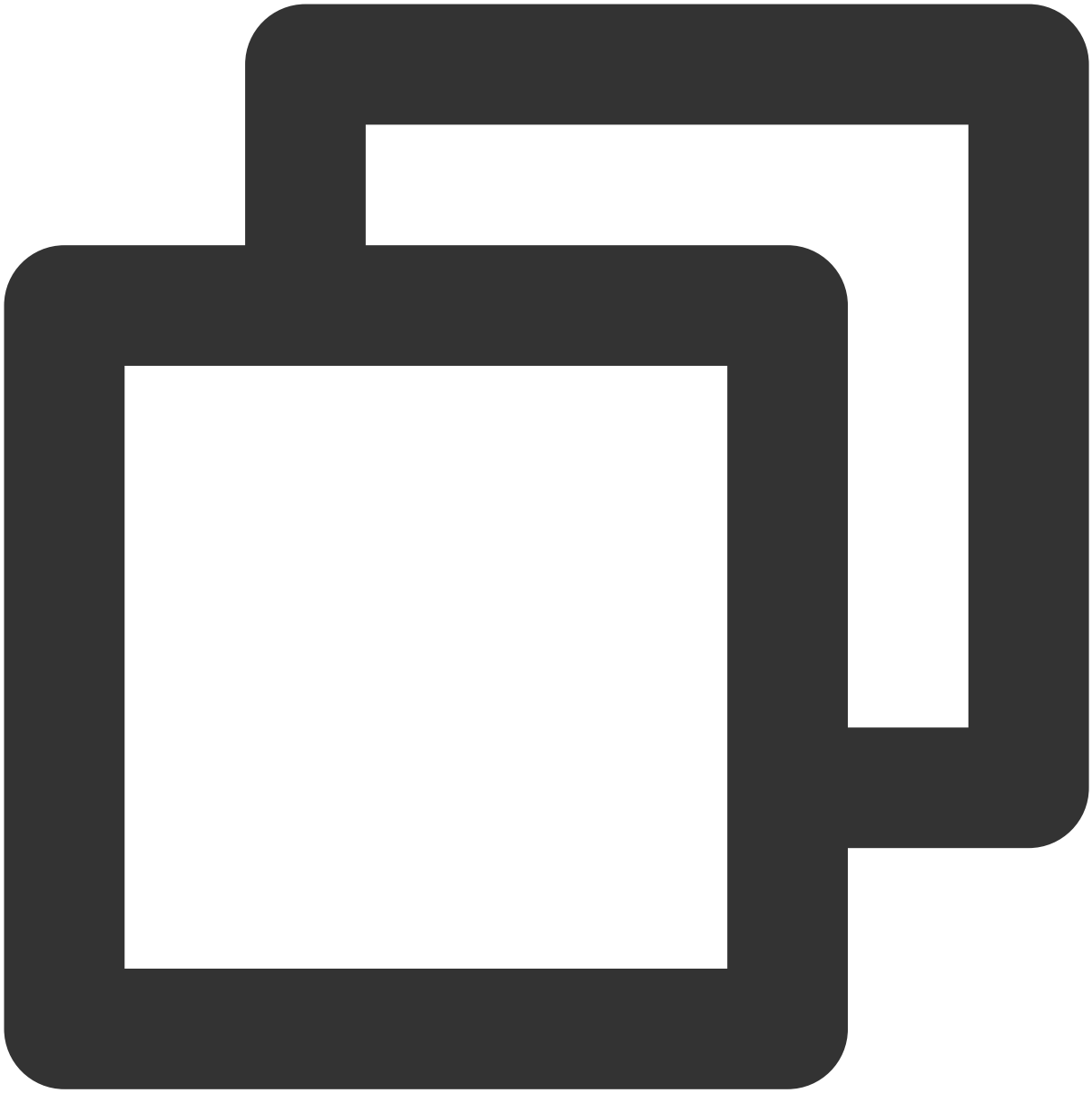
```
Base64Encode(HmacSha1("asr.cloud.tencent.com/asr/v2/1259228442?engine_model_type=16
```

The obtained `signature` value is as follows:



```
HepdTRX6u155qIPKNKC+3U0j1N0=
```

3. After the value of `signature` is **URL-encoded (URL-encoding is required; otherwise, authentication may fail**), the final request URL obtained through concatenation is:



```
wss://asr.cloud.tencent.com/asr/v2/1259228442?engine_model_type=16k_zh&expired=1592
```

Opus audio stream encapsulation description

The size of a compressed frame should be fixed at 640, which means compressing 640 shorts at a time; otherwise, decompression will fail. The spliced frames can be passed to the server, and each frame must meet the following format requirements:

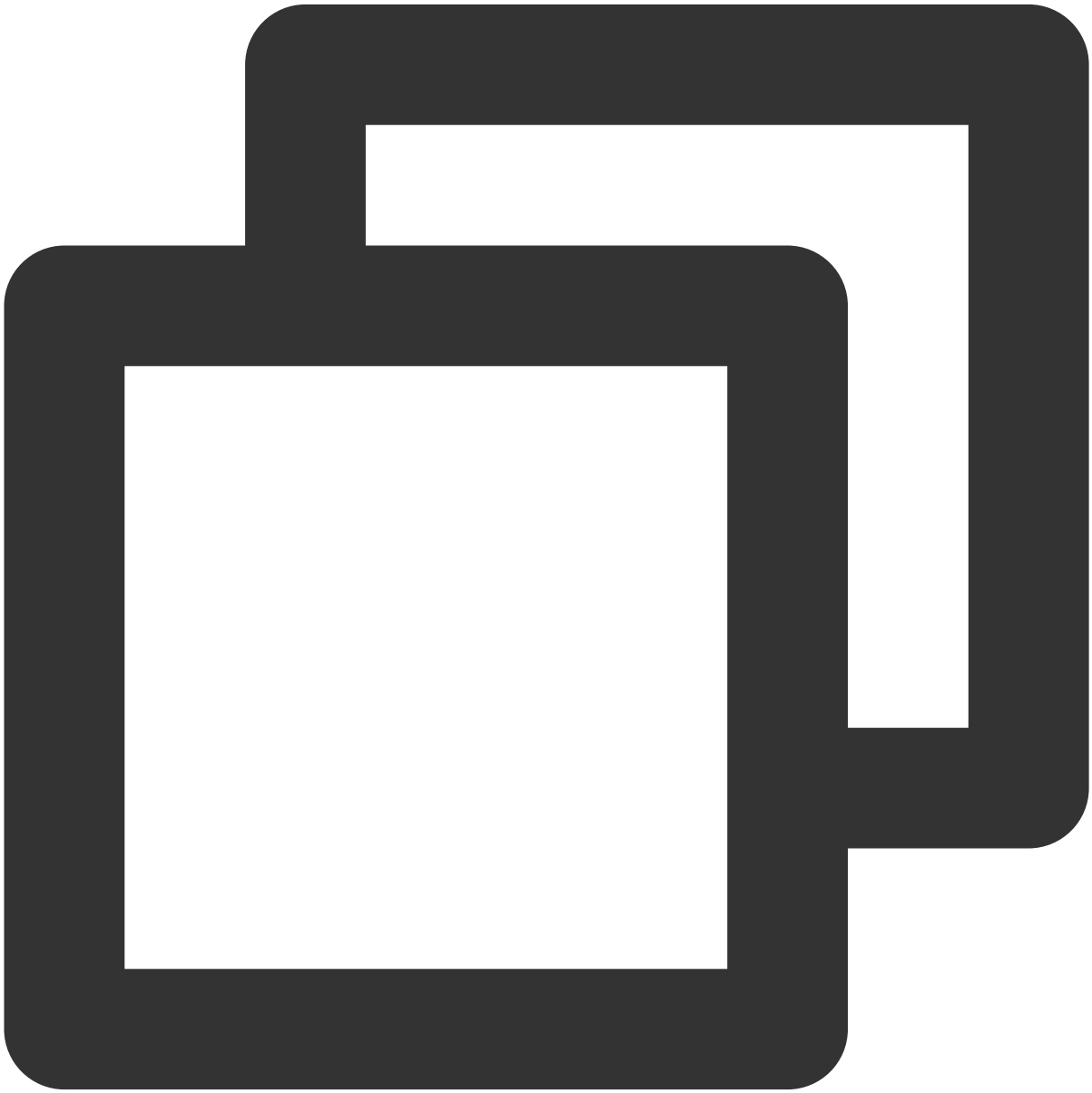
Each frame of compressed data is encapsulated as follows:

OpusHead (four bytes)	Frame data length (two bytes)	One frame of compressed Opus data

opus	Length (`len`)	Opus-decoded data in the length of `len`

Request response

After the client initiates a connection request, the backend will establish a connection and verify the signature. If the verification is passed, the backend will return an acknowledgment message with the code 0 to indicate that the handshake is successful; otherwise, the backend will return a message with a code other than `0` and close the connection.



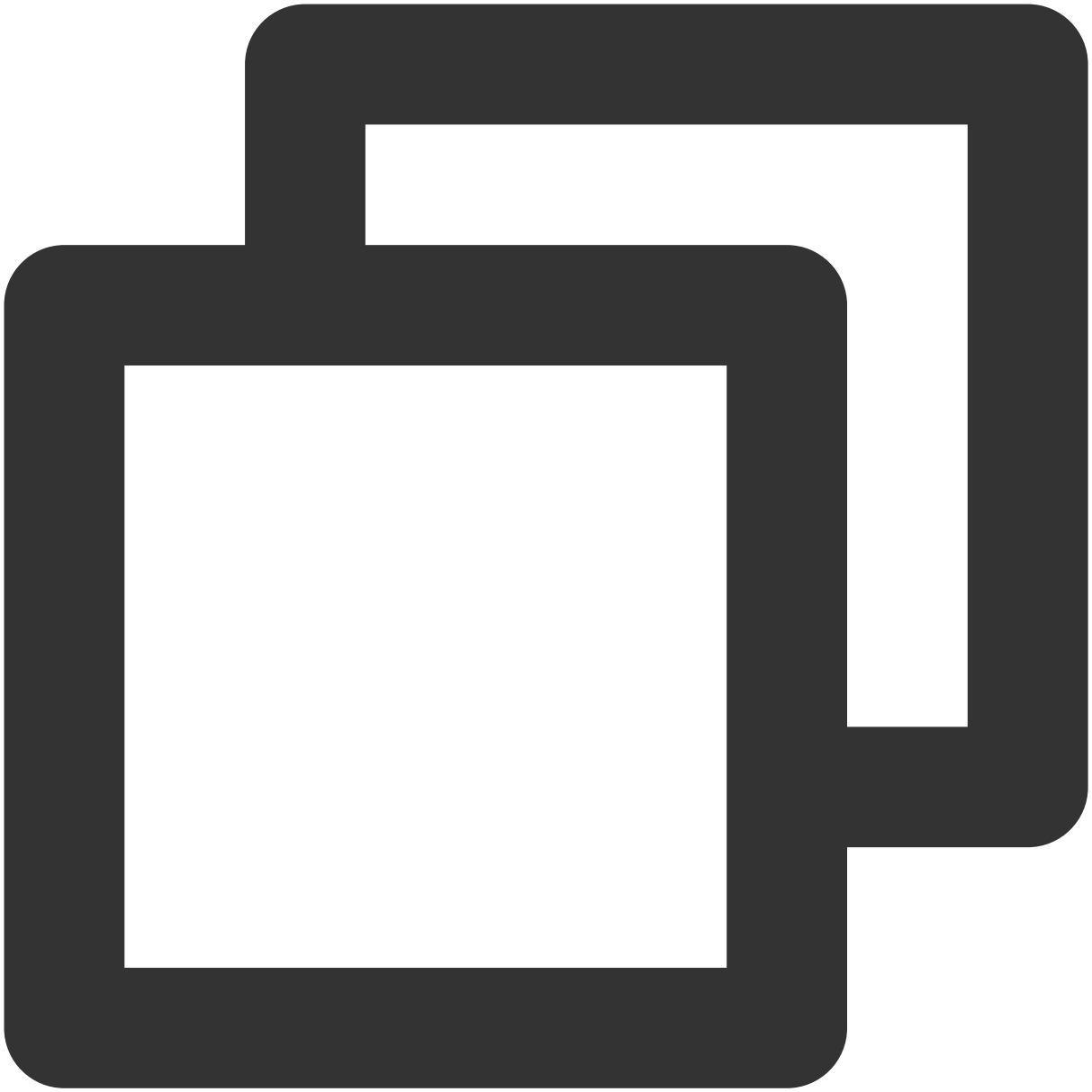
```
{"code":0,"message":"success","voice_id":"RnKu9FODFHK5FPpsrN"}
```

Recognition phase

After the handshake is successful, the backend will proceed to the recognition phase, where the client uploads audio data and receives the recognition result messages.

Uploading data

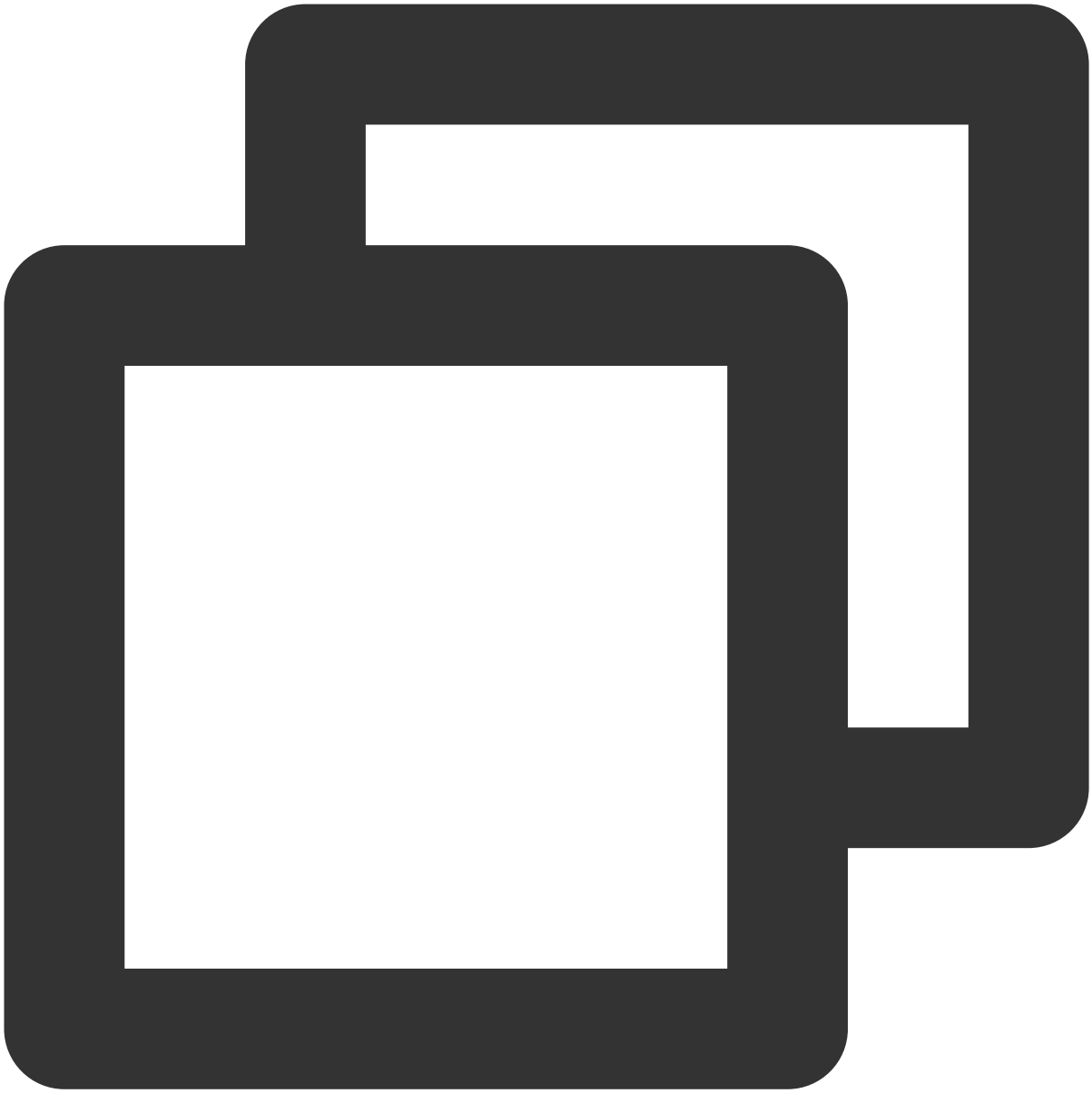
During recognition, the client continuously uploads binary messages containing binary audio stream data to the backend. We recommend you send a packet of 40 ms in length every 40 ms (i.e., real-time factor (RTF) of 1:1). The corresponding PCM is 640 bytes at a sample rate of 8 kHz or 1,280 bytes at a sample rate of 16 kHz. An audio sending rate exceeding the 1:1 RTF or an audio packet sending interval exceeding 6 seconds may cause an error in the engine. In this case, the backend will return an error and close the connection.
After the audio stream is uploaded, the client needs to send the following text message to the backend to end the recognition.



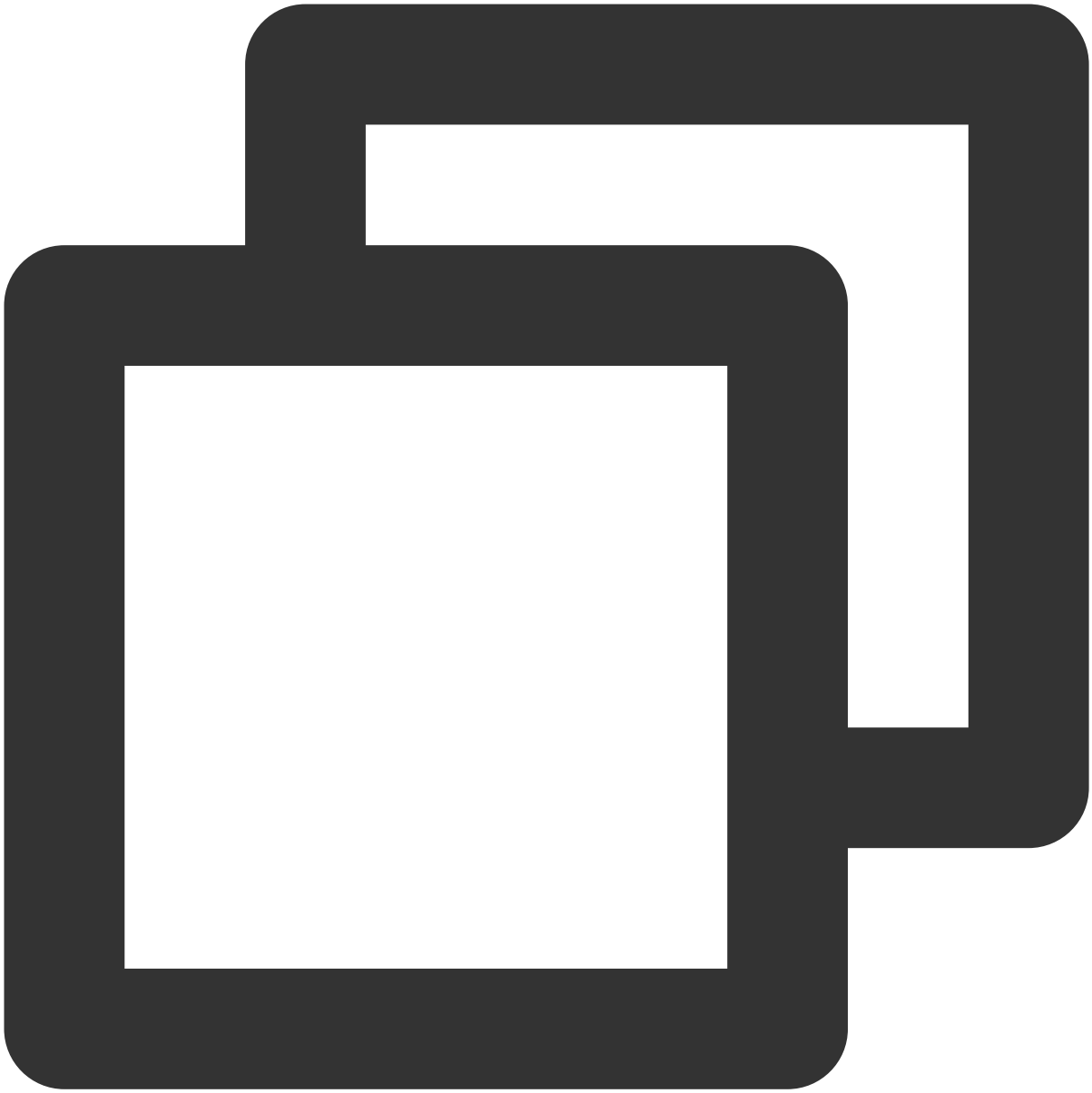
```
{"type": "end"}
```

Receiving a message

While uploading data, the client needs to receive the real-time recognition result returned by the backend synchronously. Below is a sample result:

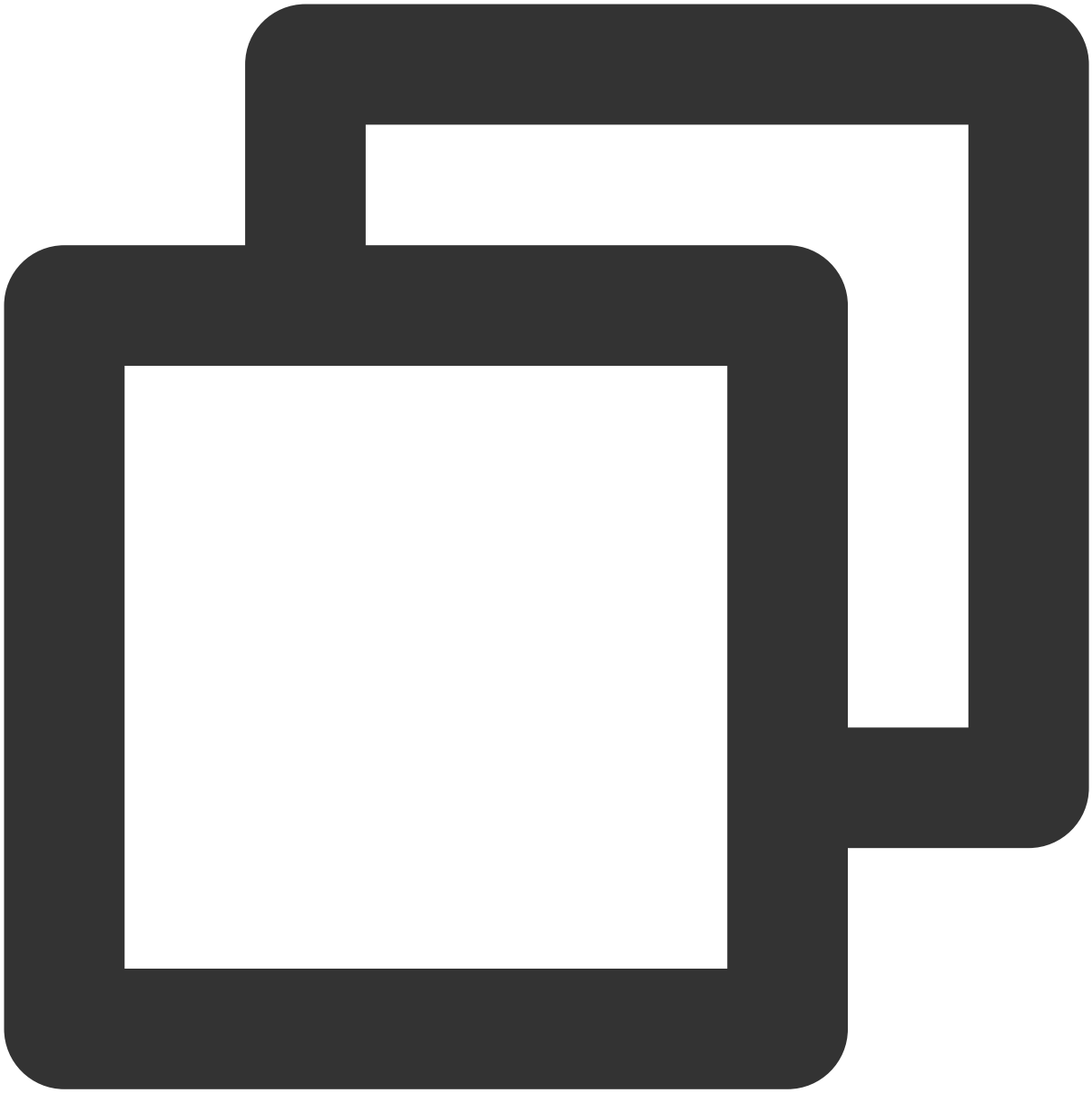


{"code":0,"message":"success","voice_id":"RnKu9FODFHK5FPpsrN","message_id":"RnKu9F



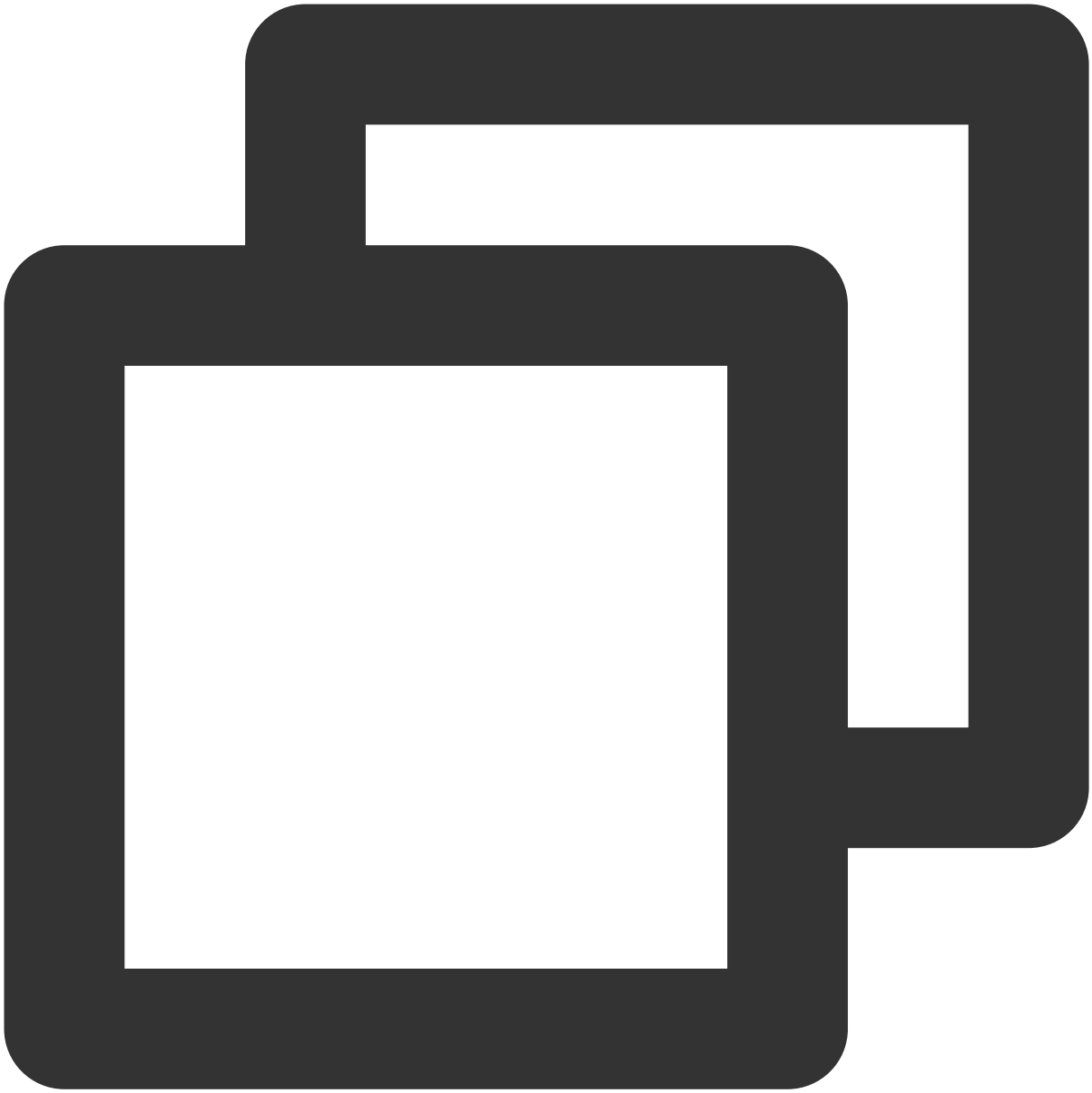
```
{"code":0,"message":"success","voice_id":"RnKu9FODFHK5FPpsrN","message_id":"RnKu9FO
```

After the backend recognizes all the uploaded audio data, it will return a message with the value of `final` being `1` and close the connection.



```
{"code":0,"message":"success","voice_id":"CzhjnqBkv8lk5pRUxhpX","message_id":"Czhjn
```

If an error occurs during recognition, the backend will return a message with a code other than 0 and close the connection.



```
{"code":4008,"message":"The backend recognition server timed out while waiting for
```

Developer Resources

SDK

Tencent Cloud Speech SDK for Go

Tencent Cloud Speech SDK for Java

Tencent Cloud Speech SDK for C++
Tencent Cloud Speech SDK for Python
Tencent Cloud Speech SDK for JS

Sample SDK call

Sample for Go
Sample for Java
Sample for C++
Sample for Python
Sample for JS

Error Codes

Value	Description
4001	The parameter is invalid. For more information, see the `message` .
4002	Authentication failed.
4003	The `AppID` has not activated the service. Activate the service in the console first.
4004	No free tier is available.
4005	The service has been interrupted due to overdue payments under your account. Top up your account in time.
4006	The current number of concurrent calls under the account exceeded the limit.
4007	Audio decoding failed. Check whether the format of the uploaded audio data is consistent with the call parameter.
4008	The client timed out while uploading data.
4009	The client was disconnected.
4010	An unknown text message was uploaded from the client.
5000	A backend error occurred. Try again.
5001	Recognition failed on the backend recognition server. Try again.
5002	Recognition failed on the backend recognition server. Try again.