API reference

WebSocket API

Real-time transcription over WebSocket

Soniox Speech-to-Text WebSocket API enables low-latency transcription of live audio streams. It supports advanced features such as automatic speaker diarization, context customization, and more — all over a persistent WebSocket connection.

This API is ideal for live transcription scenarios such as meetings, broadcasts, voice interfaces, and real-time voice applications.

WebSocket endpoint

To connect to the WebSocket API, use:

```
wss://stt-rt.soniox.com/transcribe-websocket
```

Authentication and configuration

Before sending audio, you must authenticate and configure the transcription session by sending a JSON message like this:

```
{
  "api_key": "<SONIOX_API_KEY|SONIOX_TEMPORARY_API_KEY>",
  "model": "stt-rt-preview",
  "audio_format": "auto"
}
```

Configuration parameters

`api_key` Required `string`

Your Soniox API key. You can create keys in the Soniox Console. For client-side integrations, use a temporary API key generated on the server to avoid exposing secrets.

`model` Required `string`

The transcription model to use. Use GET /models endpoint to retrieve a list of available models.

Example: `"stt-rt-preview"`

`audio_format` Required `string`

The format of the streamed audio. See Supported audio formats for details.

Example: `"auto"`, `"pcm_s16le"`

`num_channels` `number`

Required for raw PCM formats.

Common values: `1` for mono audio, `2` for stereo audio

`sample_rate` `number`

Required for raw PCM formats.

Common value: `16000`

`language_hints` `array<string>`

Expected languages in the audio. If not specified, languages are automatically detected. See supported languages for list of available ISO language codes.

`context` `string`

Provide domain-specific terms or phrases to improve recognition accuracy.

Maximum length: `10000`

`enable_speaker_diarization` `b`

When `true`, speakers are identified and separated in the transcription output.

`enable_non_final_tokens` boolean

When `true`, partial non-final tokens will be streamed before they are finalized. See Final vs non-final tokens for more information.

Default: `true`

`max_non_final_tokens_duration_ms` number

Maximum delay (in milliseconds) between a spoken word and its finalization.

Default: `4000`

Minimum: `360`

Maximum: `6000`

`enable_endpoint_detection` boolean

When `true`, endpoint detection is enabled.

`translation` object

Configure real-time translation. See Real-time transcription page for more info.

`target_language` Required string

The target language for translation. Required if `translation` is set.

`source_languages` Required array<string>

List of source languages to translate. Use `["*"]` to include all.

`exclude_source_languages` array<string>

Languages to exclude from translation. Only allowed when `source_languages` is `["*"]`.

`two_way_target_language` string

Enables two-way translation for conversations. All speech is translated between the two languages. Cannot be used with `exclude_source_languages`.

`client_reference_id` `string`

Optional tracking identifier string. Does not need to be unique.

Maximum length: `256`

Audio streaming

After sending the initial configuration, begin streaming audio data:

- Audio can be sent as binary WebSocket frames (preferred)
- Alternatively, Base64-encoded audio can be sent as text messages (if binary is not supported)
- The maximum duration of a stream is 65 minutes

Ending the stream

To gracefully end a transcription session:

- Send an empty WebSocket message (empty binary or text frame)
- The server will return any final results, send a completion message, and close the connection

---

Response format

Soniox will send transcription responses in JSON format. Successful transcription responses follow this format:

```
{
  "tokens": [
    {
      "text": "Hello",
      "start_ms": 600,
      "end_ms": 760,
      "confidence": 0.97,
      "is_final": true,
```

```
      "speaker": "1",
      "is_audio_event": false
    }
  ],
  "final_audio_proc_ms": 760,
  "total_audio_proc_ms": 880
}
```

Field descriptions

`tokens` `array<object>`

The list of transcribed tokens (words or subwords)

Each token may include:

`text` `string`

Token text.

`start_ms` Optional `number`

Start timestamp of the token (in milliseconds). Not included if `translation_status` is `translation`.

`end_ms` Optional `number`

End timestamp of the token (in milliseconds). Not included if `translation_status` is `translation`.

`confidence` `number`

Confidence score (`0.0` – `1.0`).

`is_final` `boolean`

Whether the token is finalized.

`speaker` Optional `string`

Speaker label (if diarization enabled).

`translation_status` Optional string

Status of the translation. Included if `translation` is configured. The value will be `"none"` if the current token will not be translated.

Possible values: `"original" | "translation" | "none"`

`language` Optional string

Language of the transcription. Included if `translation` is configured.

`source_language` Optional string

Source language of the translation. Included if `translation` is configured and `translation_status` is `translation`.

`final_audio_proc_ms` number

Amount of audio processed and finalized (in ms)

`total_audio_proc_ms` number

Total audio processed (in ms), including non-final tokens

Finished response

At the end of the stream, Soniox will send a final message indicating the session is complete:

```
{
  "tokens": [],
  "final_audio_proc_ms": 1560,
  "total_audio_proc_ms": 1680,
  "finished": true
}
```

The server will then close the WebSocket connection.

Error response

If an error occurs, the server will send an error response and immediately close the connection:

```
{
  "tokens": [],
  "error_code": 503,
  "error_message": "Service is currently overloaded. Please retry your request..."
}
```

`error_code` — `number`

Standard HTTP status code.

`error_message` — `string`

A description of the error encountered.

Possible error codes and their descriptions:

- 400 Bad request
- 401 Unauthorized
- 402 Payment required
- 408 Request timeout
- 429 Too many requests
- 500 Internal server error
- 503 Service unavailable

‹ Get transcriptions GET
Retrieves list of transcriptions.

Community and support ›
Engage with our community to explore new u...