API Reference Guide

AudioCodes Speech-to-Text Transcription Engine for Voice.AI

AudioCodes Speech – LVCSR WebSocket API

Version 0.60

audiocodes

Table of Contents

Notice

Information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Updates to this document can be downloaded from https://www.audiocodes.com/library/technical-documents.

This document is subject to change without notice.

Date Published: October-22-2023

Customer Support

Customer technical support and services are provided by AudioCodes or by an authorized AudioCodes Service Partner. For more information on how to buy technical support for AudioCodes products and for contact information, please visit our website at https://www.audiocodes.com/services-support/maintenance-and-support.

Stay in the Loop with AudioCodes

Abbreviations and Terminology

Each abbreviation, unless widely used, is spelled out in full when first used.

Related Documentation

Document Name
LTRT-26007 AudioCodes Automatic Speech Recognition - WebSocket API (v0.49)
LTRT-26009 AudioCodes Speech REST API (v0.5)
LTRT-26004 AudioCodes Speech –Speaker Recognition Enrollment and Segment WebSocket API (v0.11)

Document Revision Record

LTRT	Description
26002	Initial document release for Version 059.
26008	Added two API parameters.
26010	Changes to designated detection behavior and examples updated. release for Version 060.

Documentation Feedback

AudioCodes continually strives to produce high quality documentation. If you have any comments (suggestions or errors) regarding this document, please fill out the Documentation Feedback form on our website at https://online.audiocodes.com/documentation-feedback.

1 Introduction

This document describes how to utilize AudioCodes' Large Vocabulary Continuous Speech Recognition (LVCSR) technology specifically designed for Speech To Text (STT) in various use cases for both live and offline. The technology is operated via WebSocket-based protocol API, which is explained below. The document provides details of LVCSR session interaction including the parameters that governs the recognition session.

1.1 LVCSR

During the LVCSR process, the server continuously scans the speech samples to provide a transcription of spoken words in the language specified. The server creates on-the-fly reports of spoken words with their segmentation and confidence.

LVCSR Speech Recognition web socket endpoint

```
ws[s]://<server IP>:<port>/api/v1/speech:recognizeLVCSR
```

1.2 Understanding Session and Process

For the purpose of this document, the word *"Session"* refers to a web-socket session, while the term *"Process"* refers to an AudioCodes Speech server task executed asynchronous. A process utilizes a session for communications, and once the process is over, the same session can execute another process in exactly the same way. Whenever the term *"status"* is mentioned, it refers to process status.

2 API Reference

2.1 Textual message payload to server

1. Request to start LVCSR with parameters governing the process in JSON format.
2. Request to stop LVCSR in JSON format.
3. Base64 based audio streaming.

To request an LVCSR process, send action **start** (or use **start-mc** for stereo processing over default mix down stereo to mono in the case of stereo stream) with ***context*** specifying the context to use (context needs to be present on the server and must be coordinated to the specified domain). The speech language is specified through the ***accept-language*** parameter. The type of waveform samples supplied to the server must be provided in the parameter ***content-type***.

Adhoc glossary is an optional feature of ASR/LVCSR recognition customization. it allows you to add preset words and/or phrases that should be recognized in your audio data. The adhoc glossary is passed in the session ASR /LVCSR API together with phonetic lexicon (optional), adhoc-glossary-strictness, and adhoc-glossary-sensitivity parameters.

adhoc-glossary-strictness an optional parameter that determines how flexible the glossary could be spoken (e.g., saying "Open Word" instead of "Open Microsoft Word for Windows" and prefix & suffix flexibility). Value range: 0 to 100. Where 100 restricts recognition to the exact full phrases and exact prefix or suffix used. The value selection policy is highly dependent on the ability to characterize the recognizer actual inputs in the service. Changing the parameter value employs glossary compilation.

adhoc-glossary-sensitivity an optional parameter that determines the sensitivity in holding to the glossary defined phrases/words rather than generic speech. Value range: 0 to 100. Where 100 sensitivity is more attuned to the glossaries phrases (low rate of detection but higher accuracy), and 0 leads to less recognition to glossaries phrases and allows for more generic speech (high rate of detection but lower accuracy). Changing the parameter value does not employ glossary compilation.

In addition, a precompiled glossary functionality can be deployed using adhoc-glossary-id achieved by other sessions. The adhoc-glossary-id is bound and validated against a specific language and context. In case where adhoc-glossary and/or adhoc-glossary-lexicon are provided in API, they are favored over the adhoc-glossary-id.

Spotting phrases is an optional feature of LVCSR (only) recognition customization, that defines two functionalities:

- adhoc glossary of the phrases.
- designated detection for phrases, optionally marked with different reported text (tags), when detected, and provided under ‘interpretation’ in the recognition results.

Notes:

- An additional adhoc glossary may be defined in parallel to spotting phrases. The adhoc glossary lexicon will be utilized for spotting phrases as well if such provided. Pre-compiled adhoc-glossary-id could be utilized as well.
- when the spotting phrase is being set with different tags (no common), both will be reported as one phrase was detected.

Example: request from client (start request)

```
{"action":"start","content-type":"audio/l16;rate=16000","cookie":"STT
Demo","accept-language":"he-il","context":"tsw10ut","save-
waveform":"0","confidence-threshold":"0"}
```

Example: request from client for stereo stream (start-mc request)

```
{"action":"start-mc","content-
type":"audio/l16;rate=16000;channels=2","cookie":"STT Demo","accept-
language":"he-il","context":"tsw10ut","save-waveform":"0","confidence-
threshold":"0"}
```

Example: request from client (start request) with adhoc glossary and lexicon (optional)

```
{"action":"start","content-type":"audio/l16;rate=16000","cookie":"STT
Demo","accept-language":"he-il","context":"tsw10ut","save-
waveform":"0","confidence-threshold":"0","adhoc-glossary": ["voip sbc" ,
"meeting insights" , "vocanom"],"adhoc-glossary-lexicon" : {"voip": [" v
o j s o v e R a j p i","v o j p"],"sbc" : ["e s b i s i"],"meeting" : ["m
i t i n g"],"insights" : ["i n s a j ts"],"vocanom" : ["v o k a n o m"]}
}
```

Example: request from client (start request) with adhoc glossary and strictness

```
{"action":"start","content-type":"audio/l16;rate=16000","cookie":"STT
Demo","accept-language":"he-il","context":"tsw10ut","save-
waveform":"0","confidence-threshold":"0","adhoc-glossary": ["voip sbc" ,
"meeting insights" , "vocanom"],"adhoc-glossary-strictness": 100 }
```

Example: request from client (start request) with adhoc glossary id

```
{"action":"start","content-type":"audio/l16;rate=16000","cookie":"STT
Demo","accept-language":"he-il","context":"tsw10ut","save-
waveform":"0","confidence-threshold":"0","adhoc-glossary-id": " sttg-
387c2e04d5cf4741bc6bdd7bba30aea3-
748d0485a1cc01e8d756129c6300d186147678064db798aa2a8ec0275d8abcec" }
```

Example: request from client (start request) with adhoc glossary id and sensitivity

```
{"action":"start","content-type":"audio/l16;rate=16000","cookie":"STT
Demo","accept-language":"he-il","context":"tsw10ut","save-
waveform":"0","confidence-threshold":"0","adhoc-glossary-id": " sttg-
387c2e04d5cf4741bc6bdd7bba30aea3-
748d0485a1cc01e8d756129c6300d186147678064db798aa2a8ec0275d8abcec","adhoc-
glossary-sensitivity": 50}
```

Example: request from client (start request) with spotting phrases:

```
{"action":"start","content-type":"audio/l16;rate=16000","cookie":"STT
Demo","accept-language":"he-il","context":"tsw10ut","save-
waveform":"0","confidence-threshold":"0", "spotting-phrases": [{"tag":
"Emergency" , "phrases": ["Please help me" , "there is a fire"]} ,
{"phrases": ["The tag of the phrase to detect is the phrase itself"]}]}
```

Example: request from client (start request) with adhoc glossary and lexicon and spotting phrases:

```
{"action":"start","content-type":"audio/l16;rate=16000","cookie":"STT
Demo","accept-language":"he-il","context":"tsw10ut","save-
waveform":"0","confidence-threshold":"0","adhoc-glossary": ["voip sbc" ,
"meeting insights" , "vocanom"],"adhoc-glossary-lexicon" : {"voip": [" v
o j s o v e R a j p i","v o j p"],"sbc" : ["e s b i s i"],"meeting" : ["m
i t i n g"],"insights" : ["i n s a j ts"],"vocanom" : ["v o k a n o m"]},
"spotting-phrases": [{"tag": "Emergency" , "phrases": ["Please help me" ,
"there is a fire"]} , {"phrases": ["The tag of the phrase to detect is
the phrase itself"]}] }
```

Several media types are supported via "content-type":

- audio/l16;rate=8000
- audio/l16;rate=16000
- audio/PCMA;rate=8000
- audio/PCMU;rate=8000
- audio/PCMA;rate=16000
- audio/PCMU;rate=16000

To process stereo streams, the media type must indicate the number of channels, otherwise it will be considered as a mono stream.

e.g., audio/PCMU;rate=16000**;channels=2**

optionally media type may indicate explicitly that the stream is mono by denoting channels is one, e.g., audio/PCMU;rate=16000**;channels=1**

2.2 Textual message payload from server

From time to time the server notifies the status, on-the-fly results to the client in JSON format. client should monitor the status and manage the process accordingly.

Example: LVCSR process (status report)


```
{"error":{},"name":"48de1dc3-b629-42a6-b1c9-
bd6a00d3a484","type":"audiocodes.speech.LVCSROperation","status":"READY",
"cookie":"STT Demo"}
```


Example: LVCSR process (events report)


```
{"name":"48de1dc3-b629-42a6-b1c9-
bd6a00d3a484","type":"audiocodes.speech.LVCSROperation","status":"STARTED
","cookie":"STT
Demo","events":[{"stability":0.5,"alternatives":[{"transcript":"בוקר טוב #,
מה
שלומך","confidence":0.999183,"words":[{"word":"בוקר","location":120,"durati
on":39,"confidence":0.999183,"final":1},{"word":"טוב","location":159,"dura
tion":30,"confidence":0.999816,"final":1},{"word":"#,","location":-
9,"duration":3,"confidence":0.01,"final":0},{"word":"מה","location":189,"
duration":18,"confidence":0.01,"final":0},{"word":"שלומך","location":207,"
duration":48,"confidence":0.01,"final":0}]}]}]}
```


Example: LVCSR process (events report of stereo stream with channel indication)


```
{"name":"48de1dc3-b629-42a6-b1c9-
bd6a00d3a484","type":"audiocodes.speech.LVCSROperation","status":"STARTED
","cookie":"STT
Demo","events":[{"channel":1,"stability":0.5,"alternatives":[{"transcript
":"בוקר טוב #, מה
שלומך","confidence":0.999183,"words":[{"word":"בוקר","location":120,"durati
on":39,"confidence":0.999183,"final":1},{"word":"טוב","location":159,"dura
tion":30,"confidence":0.999816,"final":1},{"word":"#,","location":-
9,"duration":3,"confidence":0.01,"final":0},{"word":"מה","location":189,"
duration":18,"confidence":0.01,"final":0},{"word":"שלומך","location":207,"
duration":48,"confidence":0.01,"final":0}]}]}]}
```


Example: LVCSR process (events report with spotted phrase)


```
{"cookie":"1-
1","events":[{"alternatives":[{"confidence":0.01,"spottedphrases":[[{"con
fidence":0.621,"duration":81,"indexes":[0,1],"interpretation":["BOKERTOV"
],"location":369,"phrase":"ובוקר
טוב"}],[{"confidence":0.844,"duration":78,"indexes":[2],"interpretation":[
"KULAM"],"location":450,"phrase":"לכולם"}]],"transcript":"ובוקר טוב לכולם
#EOS_SYMBOL","words":[{"confidence":0.621,"duration":51,"final":1,"index"
:0,"location":369,"word":"ובוקר"},{"confidence":0.999,"duration":30,"final
":1,"index":1,"location":420,"word":"טוב"},{"confidence":0.844,"duration":
78,"final":1,"index":2,"location":450,"word":"לכולם"},{"confidence":0.001,
"duration":3,"final":1,"location":-
9,"word":"#EOS_SYMBOL"}]}],"stability":0.5}],"name":"a1d553ae-a6ab-4ebb-
a31e-
5a04068417fb","status":"STARTED","type":"audiocodes.speech.LVCSROperation
","waveform-tag":""}
```

2.3 Reading and interpreting LVCSR results

As seen in the example immediately above, from time to time the server sends results for the ongoing LVCSR process.

The results are provided in JSON text which describes the results at the phrase level ("transcript") and at the word level ("word")

Words carry information specifying the location and duration of the word (x10 msec units), its confidence and its state ('final' or 'not final').

Words that are defined as **"final":0** are termed *intermediate*, these words are reported as temporary hypothesis until a final decision is reported for the time segment. Any location, duration or confidence report for these words are ignored.

You can use the non-final word reports to give a hint that some LVCSR report is being formed.

When a hypothesis is complete the words in that hypothesis appear with **"final":1**, from that moment on, the reports for the location and duration of these words is locked, any future LVCSR reports will apply to later times.

Words that carry # prefix are special metadata words carrying information about punctation, etc., the application should take care of usage according to the use case. They can be omitted initially.

In addition, intermediate words (**"final":0**) may include deletion of previous intermediate words (e.g., recognizer decides it was silence rather than the previous reported words). In such a case, non-finalized transcription word results are replaced with blank (i.e., no speech detected).

Example: LVCSR process (intermediate empty event)

```
{"cookie":"STT Demo","events":[{"alternatives":[{"confidence":0.01,
"transcript":"
","words":[{"confidence":0.001,"duration":3,"final":0,"location":-
9,"word":" " }]}],"stability":0.5}],"name":"48de1dc3-b629-42a6-b1c9-
bd6a00d3a484","status":"STARTED","type":"audiocodes.speech.LVCSROperation
","waveform-tag":""}
```

2.4 Binary message payload to server

Streaming audio in multiple chunks.

recommended chunk size – amount of bytes equivalent to 250 msec of audio according to the media type.

2.5 Text message for base64 based audio streaming to server

Streaming audio in multiple chunks utilizing base64 text audio streaming:

```
 {"action":"stream",
"text":"kdXUUpEQkpmd3ENGEgZiBhZmE2MzQ1MjQyMXRycXdzZA==")
```

Zero bytes equivalent payload in base64 based audio streaming indicates to the server to stop the recognition process.

```
 {"action":"stream", "text":""})
```

2.6 Usage of adhoc glossary id

The JSON string entity specifies the glossary ID to be used in the session. The adhoc glossary ID, obtained from the response as 'adhoc-glossary-id-tag', is utilized when an adhoc glossary is employed, with or without an adhoc glossary lexicon. The purpose of this approach is to enable the use of pre-compiled glossary functionalities without explicitly providing the adhoc-glossary and lexicon, if they are provided.

2.7 Format of adhoc glossary

A JSON entity providing a string array of where its elements designate the word or phrases of the glossaries.

Example:

```
"adhoc-glossary": ["voip sbc" , "meeting insights" , "vocanom"]
```

2.8 Format of lexicon for providing transcriptions to adhoc glossary

The JSON entity contains a mapping of "attribute" to "value" pairs. The "**attribute**" represents the word's spelling, while the "**value**" is an array consisting of one or more transcriptions for that word. Linguistic support is necessary for word transcriptions. Although the adhoc glossary lexicon is not obligatory, it is recommended as it enhances accuracy.

Example:

```
"adhoc-glossary-lexicon":{"מיה":["m i a","m i j a"],"תודה":["t o d a"]}
```

2.9 Format of spotting phrases

A JSON entity provides a string array of where its elements designate word or phrases of the phrases to be detected as well as adhoc glossary.

Example:

```
"spotting-phrases": [{"tag": "Emergency" , "phrases": ["Please help me" ,
"there is a fire"]} , {"phrases": ["The tag of the phrase to detect is
the phrase itself"]}]
```

2.10 Session Termination

The process can be stopped at any time by either method according to the case:

- Application working from files for offline use:
  - Issue zero bytes binary, or
  - Issue empty text json “text” property in base64 text-based audio streaming text message
- Server to digest whatever it accumulated and not processed yet - Send "stop" request.
- Server to stop immediately and do not digest whatever it accumulated and not processed yet - Send "abort" request.

Example request from client (stop request)

```
{"action":"stop"," cookie":" STT Demo "}
```

After the “ABORTED” status is received (possibly accompanied with last events), the websocket may be closed.

2.11 Meaning and function of various LVCSR parameters

Listed below are a few prominent LVCS recognition parameters:

confidence-threshold

This parameter threshold is employed in deciding which results to transfer to the client, a sentence with confidence value lower than confidence-threshold is removed from the results. The range of this parameter is 0-1.

In LVCSR tasks, it is recommended to keep it = 0.

save-waveform

The parameter flag is used to indicate whether the received waveform should be saved in its original form as provided by the server. The server includes a waveform-tag in the reported messages indicating the recording location.

Set this flag to 1 - indicating to save the audio waveform received by the server.

2.11.1 Utilizing waveform tag

- The waveform tag can be used to operate with the session recording audio file from the server using the following REST endpoint:

```
http[s]://<server IP>:<port>/v1/<waveform-tag>
```

- HTTP methods
  - GET – downloads the audio file.
    - HTTP status codes
      - 200 – ok (file found and attached)
  - DELETE – delete the audio file and textual log file.
    - HTTP status codes
      - 200 – ok
      - 404 – not found (no audio or textual log file present)
      - 500 – server internal error (e.g., file is open, no permission, etc.)

In addition to the parameters, the server also allows a string value to be passed along with text messages (this is called ***cookie***). The ***cookie*** is present in all subsequent server text messages resulting from the process **start** message. You can use it to identify the particular process within multiple asynchronous processes.

2.12 Session Status Transitions

As can be observed from the examples above, the server reports process status within each text message. The status starts as **READY** right after the process is initialized, it continues with status **STARTED** until either a process has failure (some error occurred) in which case the server changes the status to **FAILED**, or the process has ended.

It is the responsibility of the client to stop the LVCSR process, this is described above in Session Termination.

The server stops the process upon receiving a **stop** request and is completed in the following stages:

1. Server stops reading samples from binary messages.
2. If "**stop**" has been issued, the server processes all samples that have been already received and accumulated, at the same time any events are issued as it is during real-time.
3. Once all samples are processed the process status changes to **TERMINATED** and the server emits the final events (if there are any). This is why after the status is reported to be **TERMINATED** you can still see events generated. these events will report words that have the **final** flag 1.

Example:


```
{"error":{},"cause":"success","name":"48de1dc3-b629-42a6-b1c9-
bd6a00d3a484","type":"audiocodes.speech.LVCSROperation","status":"TERMINA
TED","cookie":"STT Demo","response":{"alternatives":[]}}
```


4. When the process is halted completely the status becomes **ABORTED**

Example:


```
{"name":"48de1dc3-b629-42a6-b1c9-
bd6a00d3a484","type":"audiocodes.speech.LVCSROperation","status":"ABORTED
","cookie":"STT
Demo","events":[{"stability":0.5,"alternatives":[{"transcript":"#, מה שלומך
היום","confidence":0.01,"words":[{"word":"#,","location":-
9,"duration":3,"confidence":0.01,"final":1},{"word":"מה","location":189,"
duration":18,"confidence":0.99999,"final":1},{"word":"שלומך","location":20
7,"duration":48,"confidence":0.999847,"final":1},{"word":"היום","location"
:255,"duration":48,"confidence":0.999449,"final":1}]}]}]}
```


Example: with adhoc glossary id tag in results


```
{"name":"48de1dc3-b629-42a6-b1c9-
bd6a00d3a484","type":"audiocodes.speech.LVCSROperation","status":"ABORTED
","cookie":"STT
Demo","events":[{"stability":0.5,"alternatives":[{"transcript":"#, מה שלומך
היום","confidence":0.01,"words":[{"word":"#,","location":-
9,"duration":3,"confidence":0.01,"final":1},{"word":"מה","location":189,"
duration":18,"confidence":0.99999,"final":1},{"word":"שלומך","location":20
7,"duration":48,"confidence":0.999847,"final":1},{"word":"היום","location"
:255,"duration":48,"confidence":0.999449,"final":1}]}]}]","adhoc-
glossary-id-tag": "languages/he-il/glossaries/sttg-
387c2e04d5cf4741bc6bdd7bba30aea3-
748d0485a1cc01e8d756129c6300d186147678064db798aa2a8ec0275d8abcec"}
```


5. If "**abort**" has been issued, the server does not process samples any further, all accumulated samples are discarded.

2.13 Termination of session caused by server-side

When the session does not meet the licensing policy, the recognizer will issue an event with status “END_DETECTED”. The application stops the operation by issuing stop (either by “stop” or zero buffer)

After the “ABORTED” status is received (possibly accompanied with the last events), the WebSocket may be closed.

International Headquarters
1 Hayarden Street,
Airport City
Lod 7019900, Israel
Tel: +972-3-976-4000
Fax: +972-3-976-4040

AudioCodes Inc.
80 Kingsbridge Rd
Piscataway, NJ 08854, USA
Tel: +1-732-469-0880
Fax: +1-732-469-2298

Contact us: https://www.audiocodes.com/corporate/offices-worldwide
Website: https://www.audiocodes.com

Document #: LTRT-26010