Reference Guide

AudioCodes Intuitive Human Communications for Chatbot Services

Voice.AI Gateway Bot API

Release 1.0

ac 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: March-08-2020

WEEE EU Directive

Pursuant to the WEEE EU Directive, electronic and electrical waste must not be disposed of with unsorted waste. Please contact your local recycling authority for disposal of this product.

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
Voice.AI Gateway Product Description
Voice.AI Gateway Integration Guide

General Notes

> ⚠️ **Note:** OPEN SOURCE SOFTWARE. Portions of the software may be open source software and may be governed by and distributed under open source licenses, such as the terms of the GNU General Public License (GPL), the terms of the Lesser General Public License (LGPL), BSD and LDAP, which terms are located at https://www.audiocodes.com/services-support/open-source/ and all are incorporated herein by reference. If any open source software is provided in object code, and its accompanying license requires that it be provided in source code as well, Buyer may receive such source code by contacting AudioCodes, by following the instructions available on AudioCodes website.

Document Revision Record

LTRT	Description
30940	Initial document release.

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

AudioCodes Voice.AI Gateway enhances chatbot functionality by allowing human communication with chatbots through **voice** (voicebot), offering an audio-centric user experience. Integrating the Voice.AI Gateway into your chatbot environment provides you with a single-vendor solution, assisting you in migrating your text-based chatbot experience into a voice-based chatbot.

AudioCodes Voice.AI Gateway provides a generic bot API that can be used for connecting it to any bot service that doesn't use the standard bot frameworks (such as Microsoft Azure, Amazon Lex, and Google Dialogflow). This Customer-proprietary bot service could also employ middleware that proxies between it and the Voice.AI Gateway. In such a scenario, it's preferable that the Voice.AI Gateway connects directly to your framework or middleware.

AudioCodes bot API offers the following benefits:

- Easy to implement
- Simple authentication scheme
- Traverses firewalls and HTTP proxies
- Enables the bot to use the Voice.AI Gateway's wide range of features

> **Note:** Prior to reading this document, it's recommended that you read the Voice.AI Gateway Product Description to familiarize yourself with AudioCodes Voice.AI Gateway architecture and solution.

1.1 Purpose

This guide provides AudioCodes' APIs for connecting your bot service (proprietary bot or middleware) to AudioCodes Voice.AI Gateway.

1.2 Targeted Audience

This guide is intended for developers of bot frameworks and middleware.

This page is intentionally left blank.

2   Overview

- Roles in the bot API:
    - Client: Voice. AI Gateway
    - Server: Your bot service
- You should implement the server-side of the API so that the Voice.AI Gateway can connect to it.
- The API uses HTTP. All requests by the Voice.AI Gateway are sent to the bot service.
- The API only conveys textual messages (not voice), as the Voice.AI Gateway uses speech-to-Text (STT) and Text-to-Speech (TTS) engines.

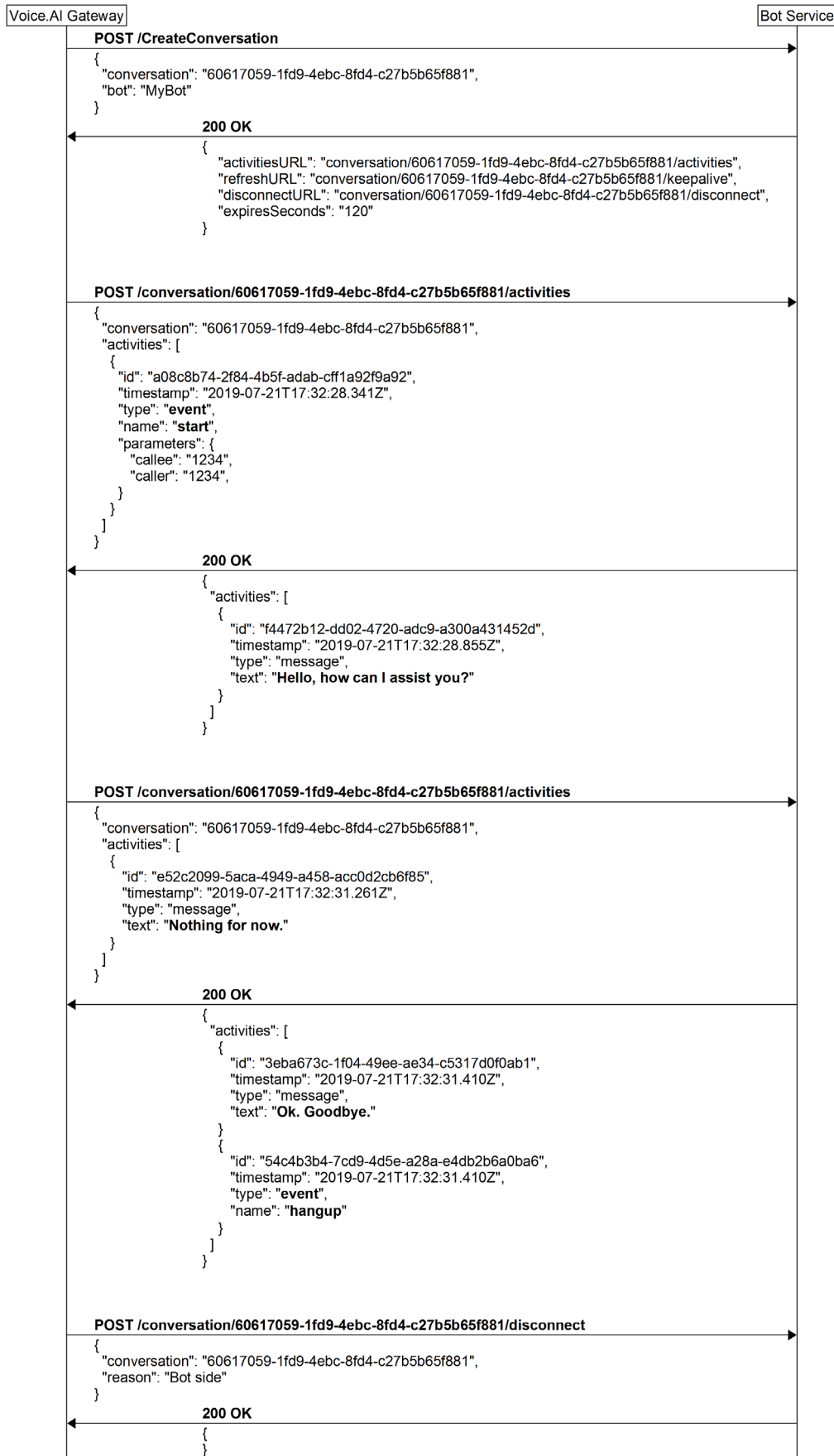**This page is intentionally left blank.**

3   Conversation Flow

The conversation flow between the Voice. AI Gateway and the bot service is as follows:

1. The Voice.AI Gateway creates a new conversation by using a pre-configured URL.

2. The reply contains URLs for posting messages to the conversation.

3. Throughout the conversation, the Voice.AI Gateway posts the user's messages to the given URL, while the responses contains the bot's replies.

4. The Voice.AI Gateway ends the conversation.

The following shows an example of a conversation flow between the Voice.AI Gateway and a proprietary bot service:

Figure 3-1: Example of Conversation Flow between Voice.AI Gateway and Bot Service

Voice.AI Gateway   Bot Service

POST /CreateConversation

```
{
  "conversation": "60617059-1fd9-4ebc-8fd4-c27b5b65f881",
  "bot": "MyBot"
}
```

200 OK

```
{
  "activitiesURL": "conversation/60617059-1fd9-4ebc-8fd4-c27b5b65f881/activities",
  "refreshURL": "conversation/60617059-1fd9-4ebc-8fd4-c27b5b65f881/keepalive",
  "disconnectURL": "conversation/60617059-1fd9-4ebc-8fd4-c27b5b65f881/disconnect",
  "expiresSeconds": "120"
}
```

POST /conversation/60617059-1fd9-4ebc-8fd4-c27b5b65f881/activities

```
{
  "conversation": "60617059-1fd9-4ebc-8fd4-c27b5b65f881",
  "activities": [
    {
      "id": "a08c8b74-2f84-4b5f-adab-cff1a92f9a92",
      "timestamp": "2019-07-21T17:32:28.341Z",
      "type": "event",
      "name": "start",
      "parameters": {
        "callee": "1234",
        "caller": "1234",
      }
    }
  ]
}
```

200 OK

```
{
  "activities": [
    {
      "id": "f4472b12-dd02-4720-adc9-a300a431452d",
      "timestamp": "2019-07-21T17:32:28.855Z",
      "type": "message",
      "text": "Hello, how can I assist you?"
    }
  ]
}
```

POST /conversation/60617059-1fd9-4ebc-8fd4-c27b5b65f881/activities

```
{
  "conversation": "60617059-1fd9-4ebc-8fd4-c27b5b65f881",
  "activities": [
    {
      "id": "e52c2099-5aca-4949-a458-acc0d2cb6f85",
      "timestamp": "2019-07-21T17:32:31.261Z",
      "type": "message",
      "text": "Nothing for now."
    }
  ]
}
```

200 OK

```
{
  "activities": [
    {
      "id": "3eba673c-1f04-49ee-ae34-c5317d0f0ab1",
      "timestamp": "2019-07-21T17:32:31.410Z",
      "type": "message",
      "text": "Ok. Goodbye."
    }
    {
      "id": "54c4b3b4-7cd9-4d5e-a28a-e4db2b6a0ba6",
      "timestamp": "2019-07-21T17:32:31.410Z",
      "type": "event",
      "name": "hangup"
    }
  ]
}
```

POST /conversation/60617059-1fd9-4ebc-8fd4-c27b5b65f881/disconnect

```
{
  "conversation": "60617059-1fd9-4ebc-8fd4-c27b5b65f881",
  "reason": "Bot side"
}
```

200 OK

```
{
}
```

4   API

4.1   Before You Begin

Prior to using this API, please note the following:

- All Voice.AI Gateway requests use HTTP POST request methods.
- All requests and responses contain a JSON body and with the appropriate 'Content-Type: application/json' header.
- All JSON bodies must be encoded with UTF-8.
- Any non-200 response is considered a failure and disconnects the conversation. Failure responses can optionally contain a JSON body with a `reason` attribute.
- All requests have a timeout of 20 seconds. If the timeout expires and no response has been received, the conversation is disconnected.
- The Voice.AI Gateway uses connection reuse (HTTP Connection Keep-Alive). It's recommended that the bot service sets the HTTP Keep-Alive time to at least 30 seconds.
- If a connection error occurs, the Voice.AI Gateway retries the request. Note that the Voice.AI Gateway ignores duplicated activity IDs and therefore, retrying is not expected to cause double handling of the activities.

4.2   Creation of a Conversation

To start a conversation, the Voice.AI Gateway sends a POST request to a specific URL (e.g., https://example.com/api/CreateConversation). You should provide the URL to AudioCodes, so that it can be configured on the Voice.AI Gateway. The Voice.AI Gateway sends the unique ID of the conversation in the `conversation` attribute. If several bots share the same URL, the Voice.AI Gateway can be configured to add a `bot` attribute to the request body.

The body of the response from the bot service should contain a set of URLs for performing actions on the newly created conversation. The URLs should be unique to the conversation, by containing a UUID as part of the path - either by using the ID from the `conversation` attribute or a UUID generated by the bot service.

> **Note:** If a URL is relative, the Voice.AI Gateway resolves the URL using the `CreateConversation` URL as the base URL (according to Section 4 of RFC 1808).

After the conversation is created, the Voice.AI Gateway sends an activity with the `start` event. For more information on the start message, refer to the [Voice.AI Gateway Integration Guide](#).

Request Body Attributes

Parameter	Type	Description
conversation	String	Voice.AI Gateway's conversation ID.
bot	String	(Optional) The value of the providerBotName configuration parameter (if exists).

Response Body Attributes

Parameter	Type	Description
disconnectURL	String	Relative or absolute URL.
refreshURL	String	Relative or absolute URL.
activitiesURL	String	Relative or absolute URL.
expiresSeconds	Number	The value can be from 60 to 3600. The recommended value is 120. For more information on conversation refreshes, see Section Conversation Refresh.

Example

The following shows an example of creating a conversation:

■   Request:

```
{
  "conversation": "ad8f59d2-4a72-4f19-ad34-e7e9b1636111"
}
```

■   Response:

```
{
  "activitiesURL": "conversation/ad8f59d2-4a72-4f19-ad34-
e7e9b1636111/activities",
  "refreshURL": "conversation/ad8f59d2-4a72-4f19-ad34-
e7e9b1636111/refresh",
  "disconnectURL": "conversation/ad8f59d2-4a72-4f19-ad34-
e7e9b1636111/disconnect",
  "expiresSeconds": 60
}
```

4.3   Sending and Receiving Activities

The messages sent between the parties of the conversation are called *activities*. When the Voice.AI Gateway has activities to send, it sends a POST request to the URL specified in `activitiesURL`. The body of the POST request includes an `activities` attribute containing an array of activities.

The body of the response should also include an `activities` attribute containing an array of activities. If no activities are needed, either the `activities` attribute is omitted or it's sent with an empty array.

If the conversation doesn't exist, the bot service should respond with a 404 Not Found.

The format of the activities is described in the Voice.AI Gateway Integration Guide. In addition, each `activity` must include the following additional attributes:

- `id`: The sender of an activity should generate a UUID (RFC 4122, v4) per activity and send it in the `id` attribute. The receiver of activities should retain a set of all the received activities IDs (in the current conversation) and ignore duplicate activities. This allows the resending of activities in case of failures, without the activities being handled twice.

- `timestamp`: The sender of an activity should add a `timestamp` attribute containing the current time. The format of the timestamp is according to RFC 3339, where the time is in UTC with 3 decimal digits for milliseconds. For example: "2019-04-23T18:25:43.511Z".

  The `timestamp` must include the creation time of the activity and must not be modified if the activity is re-sent.

  The timestamp is mainly used for logging and debugging.

Request Body Attributes

Parameter	Type	Description
`conversation`	String	Voice.AI Gateway's conversation ID.
`activities`	Array of Objects	Array of activities.

Response Body Attributes

Parameter	Type	Description
`activities`	Array of Objects	Array of activities.

Example

The following shows an example of the `start` activity that is sent by the Voice.AI Gateway when a conversation starts (using `activities` endpoint):

- Request:

```
{
  "conversation": "ad8f59d2-4a72-4f19-ad34-e7e9b1636111",
  "activities": [
    {
      "id": "ecf2d78d-ef7b-4a5e-907c-53c97cef5f97",
      "timestamp": "2020-01-26T13:03:48.745Z",
```

```
        "type": "event",
        "name": "start",
        "parameters": {
          "callee": "1234",
          "calleeHost": "10.20.30.40",
          "caller": "+123456789",
          "callerHost": "10.20.30.40"
        }
      }
    ]
}
```

■   Response:

```
{
  "activities": [
    {
      "id": "15b3d407-5161-41e7-8114-a273859c5f6d",
      "timestamp": "2020-01-26T13:03:48.748Z",
      "type": "message",
      "text": "Hi there."
    }
  ]
}
```

The following shows an example of `message` activities that correspond to speech utterances:

■   Request (to `activitiesURL`):

```
{
   "conversation":"55b77909-82d8-4355-87f1-68081f4dbb36",
   "activities":[
      {
         "id":"bc44c054-846d-490d-85e9-d3aea96b4f0f",
         "timestamp":"2019-08-20T14:09:12.251Z",
         "type":"message",
         "text":"Hi.",
         "parameters":{
            "confidence":0.6599681377410889,
            "recognitionOutput":{
               "RecognitionStatus":"Success",
               "Offset":32300000,
               "Duration":5800000,
               "NBest":[
                  {
                     "Confidence":0.6599681377410889,
                     "Lexical":"hi",
                     "ITN":"Hi",
                     "MaskedITN":"Hi",
                     "Display":"Hi."
                  },
                  {
                     "Confidence":0.3150425851345062,
```

```
                        "Lexical":"high",
                        "ITN":"high",
                        "MaskedITN":"high",
                        "Display":"high"
                    }
                ]
            }
        }
    }
  ]
}
```

◼ Response:

```
{
  "activities": [
    {
      "id": "dc4eb401-17f2-436f-80fa-b60156b8a804",
      "timestamp": "2020-01-26T13:04:00.885Z",
      "type": "message",
      "text": "How may I assist you?"
    }
  ]
}
```

4.4   Conversation Refresh

The Voice.AI Gateway refreshes the conversation by sending a refresh request to the conversation at least 30 seconds before the `expiresSeconds` value expires. The `expiresSeconds` time is activated upon the start of conversation or last refresh. The refresh is done by sending a POST request to the URL specified in `refreshURL`.

The `expiresSeconds` value can be updated by the response body.

If the bot service doesn't receive a refresh request before `expiresSeconds` value expires, it should consider the conversation as terminated (with an error condition).

If the conversation doesn't exist, the bot service should respond with a 404 Not Found.

Request Body Attributes

Parameter	Type	Description
`conversation`	String	Voice.AI Gateway's conversation ID.

Response Body Attributes

Parameter	Type	Description
`expiresSeconds`	Number	If the conversation doesn't receive a refresh, it's closed after the time specified by this parameter. The value can be from 60 to 3600. The recommended value is 120.

Example

■   Request:

```
{
  "conversation": "ad8f59d2-4a72-4f19-ad34-e7e9b1636111"
}
```

■   Response:

```
{
  "expiresSeconds": 60
}
```

4.5   Ending a Conversation

The conversation may end due to the following reasons:

■   The VoIP call has ended (loss of connection with Voice.AI Gateway, or some failure on the SIP side).

■   The bot has disconnected (using the `hangup` event, as described in the Voice.AI Gateway Integration Guide).

■   An error has occurred.

For any of the above reasons, the Voice.AI Gateway sends a POST request to the URL specified in `disconnectURL`. The body of the POST request can contain a `reason` attribute. The body of the response should be an empty JSON object. If the conversation doesn't exist, the bot service should respond with a 404 Not Found.

Note that if the conversation expires on the bot service side (i.e., no refresh was done by the Voice.AI Gateway), no explicit message is sent by the Voice.AI Gateway.

Request Body Attributes

Parameter	Type	Description
conversation	String	Voice.AI Gateway's conversation ID.
reason	String	(Optional) The reason for disconnecting the conversation (free text).

Response Body Attributes

The response body is empty.

Example

■   Request:

```
{
  "conversation": "ad8f59d2-4a72-4f19-ad34-e7e9b1636111",
  "reason": "Client Side"
}
```

■   Response:

```
{
}
```

5   Security

5.1   TLS Usage

It's recommended that the URLs of the bot service use HTTPS.

However, for testing environments, HTTP URLs can be used. In addition, the Voice.AI Gateway can be configured to accept self-signed certificates from the bot service.

5.2   Voice.AI Gateway Authentication

It's recommended that the Voice.AI Gateway be configured with the token value that is sent in the 'Authorization: Bearer <token>' header for every HTTP request. This token is used by the bot service to authenticate the Voice.AI Gateway.

For environments that don't require this authentication (e.g., when implementing an alternative authentication method), the token can be left without a value, and no 'Authorization' header will be sent.

International Headquarters

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

AudioCodes Inc.

200 Cottontail Lane
Suite A101E
Somerset NJ 08873
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-30940

![audiocodes logo]