

Tutorials

1. Overview

1.1. Transcription Features

1.2. CX AI Features

2. Getting Started

Step 1: Create an Account

[Create a free account](#) to gain immediate access to ElevateAI.

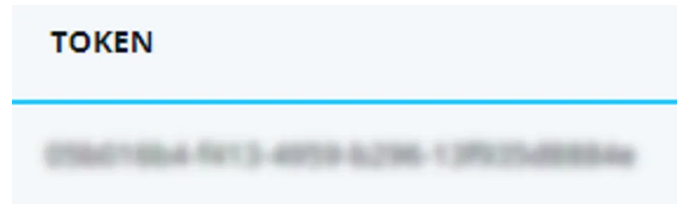
Free accounts have access to declare up to 1,000 interactions for processing per day.

If you are interested in upgrading to a paid account, you may [submit a request](#) from the ElevateAI application or contact us directly at sales@elevateai.com.

Step 2: Get Your Access Token

To enable [authentication](#) and usage tracking, an active **API token** associated with your account will be included in each request made to the ElevateAI API.

You are issued an API token immediately upon account creation.



To access your API token, simply navigate to the [Manage Tokens Dashboard](#).

Step 3: Start using ElevateAI

You're now ready to generate transcriptions, uncover CX AI insights, create autosummaries and more with ElevateAI!

Within the Interactions resource, there are **8 different endpoints** that enable you to instantly generate and retrieve insights to elevate your applications:

In general, you will need to:

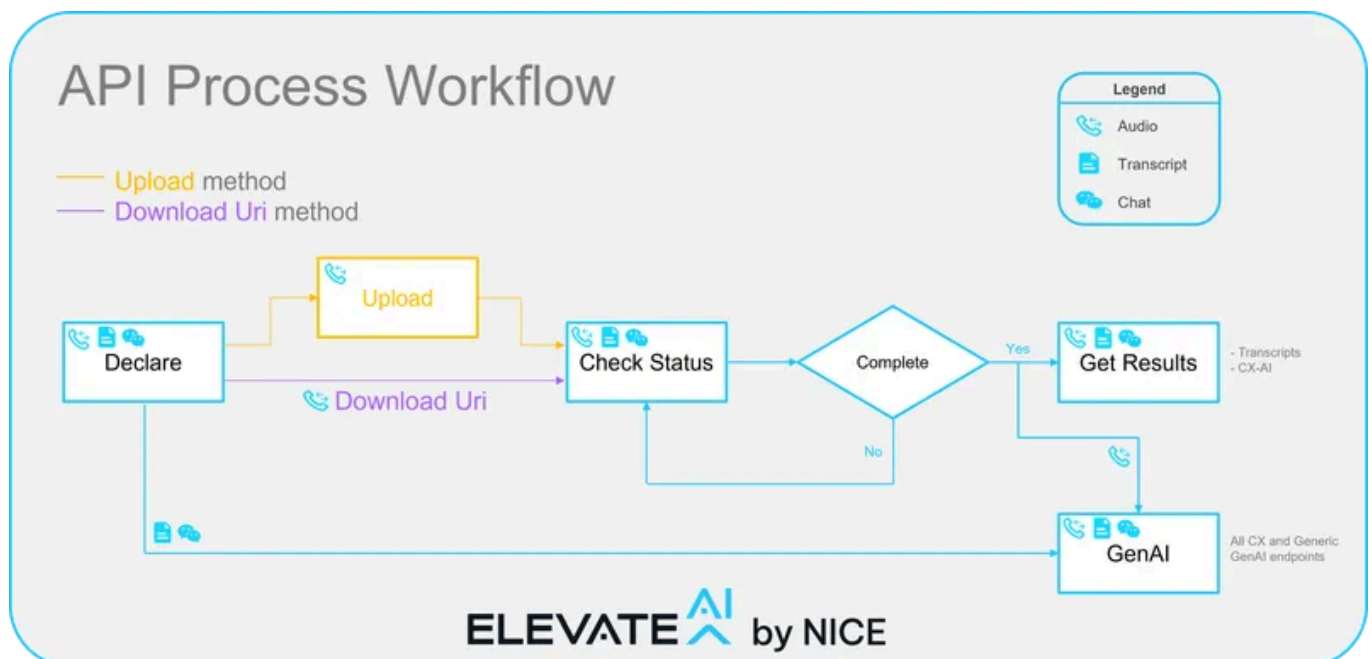
- ☐ Declare Your Media: [Declare an Audio Interaction](#), [Declare a Chat Interaction](#) or [Declare a Transcript](#)
- ☐ Check Status: [Check the Processing Status](#)
- ☐ Once Processing is Complete? Retrieve Your Results:
 - [Get CX AI](#)
 - [Get Phrase-by-Phrase Transcript](#)
 - [Get Punctuated Transcript](#)
 - [ElevateAI Generative AI API calls](#)

NOTE: With chats and transcripts, [ElevateAI Generative AI API calls](#) do not require processing to be completed

ElevateAI API Process Flow Diagram (PFD)

The following diagram depicts the API Process Flow Diagram, or Process Workflow, across the various source types available within the ElevateAI API.

NOTE: For most use cases, the calling application should use a multithreaded architecture to maximize throughput.



ElevateAI API Endpoints

The following chart lists the available ElevateAI API Endpoints. For most Generative AI endpoints there are two versions: CX and Generic; the CX Endpoints have been listed here. For additional information on the Generic Endpoints, please see [Generic Gen AI Endpoints](#).

Method	Path
POST	https://api.elevateai.com/v1/interactions Tutorial: Declare an <u>audio</u>, <u>transcript</u>, or <u>chat</u> interaction for processing
POST	https://api.elevateai.com/v1/interactions/{interactionIdentifier}/upload Tutorial: Upload a local audio file
GET	https://api.elevateai.com/v1/interactions/{interactionIdentifier} Tutorial: Retrieve declared parameters
GET	https://api.elevateai.com/v1/interactions/{interactionIdentifier}/status Tutorial: Retrieve the processing status
GET	https://api.elevateai.com/v1/interactions/{interactionIdentifier}/transcript Tutorial: Retrieve the phrase-by-phrase transcript
GET	https://api.elevateai.com/v1/interactions/{interactionIdentifier}/transcripts/punctuated Tutorial: Retrieve the punctuated transcript
GET	https://api.elevateai.com/v1/interactions/{interactionIdentifier}/ai Tutorial: Retrieve the CX AI results
GET	https://api.elevateai.com/v1/interactions/{interactionIdentifier}/gen-ai/cx/summary Tutorial: Retrieve the Gen AI Summary
GET	https://api.elevateai.com/v1/interactions/{interactionIdentifier}/gen-ai/cx/summary/details Tutorial: Retrieve the Gen AI Summary Details
GET	https://api.elevateai.com/v1/interactions/{interactionIdentifier}/gen-ai/cx/agent/action-items Tutorial: Retrieve the Gen AI Action Items
GET	https://api.elevateai.com/v1/interactions/{interactionIdentifier}/gen-ai/cx/agent/coaching-assistant Tutorial: Retrieve the Gen AI Agent Coaching Assistant
GET	https://api.elevateai.com/v1/interactions/{interactionIdentifier}/gen-ai/cx/topics

Method	Path
	Tutorial: Retrieve the Gen AI Topics
GET	https://api.elevateai.com/v1/interactions/{interactionIdentifier}/gen-ai/cx/conversation-composition Tutorial: Retrieve the Gen AI Conversation Composition
GET	https://api.elevateai.com/v1/interactions/{interactionIdentifier}/gen-ai/cx/ask Tutorial: Retrieve the Gen AI Ask ElevateAI
POST	https://api.elevateai.com/v1/interactions/{interactionIdentifier}/gen-ai/cx/ask Tutorial: Submit Question to Gen AI Ask ElevateAI
DELETE	https://api.elevateai.com/v1/interactions/{interactionIdentifier}
E	Tutorial: Delete all data

Sample Audio Files

 [Audio Sample 1.wav](#)



Not a Developer?

You can process audio files and retrieve results within the ElevateAI app with [no code](#) needed.

Ready for Production?

Contact us at sales@elevateai.com to upgrade your account today!

2.1. ElevateAI Dashboards

// Usage

The Usage Dashboard provides key insights into the **total length of files processed** for each available Microservice, as well as total processing requests declared towards your monthly request quota

All data within the dashboard and **usage export** is adjusted based on whether *Current Month*, *Previous Month*, or *Last 60 Days* is selected

Current Month ▼

Export

Overall usage within the selected timeframe is represented at the top of the dashboard for each Microservice:



Transcripti on	Total time (to the nearest hour) of all successfully processed audio interactions
Elevated Accuracy	Total time (to the nearest hour) of all successfully processed audio interactions where audioTranscriptionMode is prioritized for highAccuracy
CX AI (hours)	Total time (to the nearest hour) of all succesfully processed audio interactions where includeAiResults is defined as true
CX AI (words)	Total number of words (rounded up to the nearest 1,000 words for each interaction) of all successfully processed transcript and chat interactions

Directly under the **Usage by Date graph** is a table containing usage within the selected timeframe for each Microservice associated with each applicable token:

Name	Token	Last Request (UTC)	Requests	Transcription Hours	Elevated Accuracy Hours	CX AI Hours	CX AI Words
------	-------	--------------------	----------	---------------------	-------------------------	-------------	-------------

Name	Name associated with the token, editable in the Manage Tokens Dashboard
Token	Last 4 characters of token
Last Request (UTC)	Date/time (in UTC) of the most recent audio , transcript , or chat interaction declared with the token
Requests	Total number of audio , transcript , and chat interactions declared with the token
Transcription Hours	Total time (represented as a fraction of hours) of all successfully processed audio interactions
Elevated Accuracy Hours	Total time (represented as a fraction of hours) of all successfully processed audio interactions where audioTranscriptionMode is prioritized for highAccuracy
CX AI Hours	Total time (represented as a fraction of hours) of all successfully processed audio interactions where includeAiResults is defined as true
CX AI Words	Total number of words (rounded up to the nearest 1,000 words for each interaction) of all successfully processed transcript or chat interaction

Based on the selected time period, you can **export** usage details to track specific processing length by date and token used. The export provides additional granularity when assessing and tracking usage both overall and by date for each applicable token.

// Spend

The [Spend Dashboard](#) provides key insights into the **total billing impact** of ElevateAI usage and is only available for customers with a Paid Account.

ALI data within the dashboard adjusted dependent whether *Current Month* or *Previous Month* is selected. You can [export](#) usage details to see billing, overall and for each Microservice, for the most recent three (3) months.

See the [ElevateAI Invoicing Policy](#) for additional information.

// Active Jobs

The [Active Jobs Dashboard](#) provides key insights into the **total number of declared interactions**, by file type – **audio** , **chat** , and **transcript** – and **status** for the most recent 15 days.

Active Jobs 0	Audio 0	Chat 0	Transcript 0
Type	Status	Files	

NOTE: Interactions for which **all data has been deleted** will not appear within the Active Jobs count, but are still included in calculations impacting **Usage** and **Spend** dashboards.

// Interactions

The [Interactions Dashboard](#) enables you to **see key parameters** and **immediately download available results** from any **audio** , **transcript** , or **chat** interaction declared – including those declared directly from the application with **no code**.

Status (All) ▾	Sort By ▾	1 ▾	Interaction Guid	07/12/2022 📅	08/11/2022 📅	Go
----------------	-----------	-----	------------------	--------------	--------------	----

You are able to **filter results by status**, sort by time declared/processed, search for a specific `interactionIdentifier` , and/or filter results by a desired data range simply by selecting desired configurations at the top of the dashboard and pressing **Go**.

Identifier	Language	Type	Status	Declared (UTC)	Processed (UTC)	Error	Action
------------	----------	------	--------	----------------	-----------------	-------	--------

Table Element	Description
Identifier	Unique <code>interactionIdentifier</code> associated with request
Language	<p>Primary language of the file, will be:</p> <p><code>en-US</code> for North American English,</p> <p><code>en</code> for International English, or</p> <p><code>es-419</code> for North American Spanish</p> <p><code>pt-br</code> for Brazilian Portuguese</p> <p>Primary language of the file, will be <code>en-us</code> for North American English,</p> <p><code>en</code> for International English, or <code>es-419</code> for North American Spanish,</p> <p>Primary language of the file, will be <code>en-us</code> for North American English,</p> <p><code>en</code> for International English, or <code>es-419</code> for North American Spanish</p>
Type	Type of interaction, will be <code>Audio</code> , <code>Transcript</code> , or <code>Chat</code>
Status	Processing status of interaction
Declared (UTC)	Date/time (represented in UTC) when the <code>audio</code> , <code>transcript</code> , or <code>chat</code> interaction was successfully declared, or the date/time the <code>audio</code> interaction was successfully uploaded to the No Code dashboard
Processed (UTC)	If applicable, date/time (represented in UTC) when the interaction was successfully processed and applicable results available to to be retrieved
Error	As applicable, additional details to assist in troubleshooting
Action	<p>Click icon to automatically download results:</p> <p>Download AI</p> <p>If applicable, JSON containing detailed CX AI output</p> <p>Download Text Transcript</p> <p>If applicable, text file where each line represents a <code>sentenceSegment</code> object from the punctuated transcript</p> <p>Download Text Transcript No Speaker Info</p> <p>If applicable, text file (without speaker labels) where each line represents the start and stop time (in milliseconds) of the phrase along with the</p>

Table Element	Description
	series of words spoken without pauses
	Download JSON Transcript If applicable, JSON containing the phrase-by-phrase transcript

If desired, additional details on the [declared parameters](#), including whether an `audio` file was processed for `highSpeed` or `highAccuracy`, and detailed [processing status](#) can be requested from the API.

NOTE: Interactions for which all data has been [deleted](#) will not appear within the Interactions dashboard, but will be represented in [Usage](#) and [Spend](#) dashboards.

// No Code

The [No Code Dashboard](#) enables you to **upload** an `audio` file (or leverage one of our [sample audio files](#)) for transcription and/or CX AI analysis and **immediately download available results**.



(Upload a File)

To begin, simply click the [blue upload button](#) to open the Upload Audio Interaction widget.

The Upload Audio Interaction widget enables you to declare any [valid](#) audio file for transcription, as well as define key parameters for analysis:

Upload Audio Interaction

Select File (200 MB max)

Choose File

No file chosen

Language

en-US

Transcription Mode

HighAccuracy

Include AI Results

☒

Close

Upload

Language	Primary language of the audio file, will be: "en-US" (for North American English), "en" (for International English), or "es-419" (for North American Spanish)
Transcription Mode	Desired transcription prioritization, will be "HighSpeed" or "HighAccuracy"
Include AI Results	Check to include CX AI results

Note: Transcription requests where Audio Transcription Mode is set to [HighAccuracy](#) and/or Include AI Results is set to **true** may incur additional fees.

See the ElevateAI [Invoicing Policy](#) for additional information.

Once the Upload Audio Interaction Widget indicates that the upload is finished , you may select another interaction for processing or close the widget. The interaction(s) will appear at the top of the table.

Pro Tip: Refresh the page to see the latest [status](#).

You are able to filter results by [status](#), sort by time declared/processed, search for a specific `interactionIdentifier` , and/or filter results by a desired date range simply by selecting desired configurations at the top of the dashboard and pressing **Go**.





The table format aligns with the [Interactions Dashboard](#), and will only contain interactions declared directly from the No Code Dashboard.

// Manage Tokens

Within the [Manage Tokens Dashboard](#) you are able to issue additional tokens as well as edit the name of a token, activate a token, deactivate a token, and revoke issued tokens.

All issued tokens must be activated prior to use. To activate your token, click on the **red power icon** located under **Actions**.

ACTIONS



NAME	TOKEN	LAST REQUEST (UTC)	REQUESTS	STATUS ⓘ	ACTIONS
------	-------	--------------------	----------	----------	---------

Token Status	Description
Issued	Token is created but has not been approved by ElevateAI for use
Activated	Token is currently usable
Deactivated	Token is not currently usable, but can be reactivated at any time
Revoked	Token cannot be used and cannot be reactivated

Note: Free accounts include access to **one** API token. Paid accounts may have up to **five** concurrent API tokens.

3. ElevateAI Postman Collection

Introduction

We're excited to introduce our Postman collection for ElevateAI, designed to make exploring and integrating our API easier than ever!

This collection is a set of ready-to-use API endpoints that you can access, test, and modify directly in Postman, saving you time and effort during testing and development. Whether you're testing functionality, learning about API capabilities, or building your application, this collection provides a convenient, user-friendly way to interact with our API.

To use, simply import the collection into Postman, set your API token to the current value for the *MyApiToken* variable within the Postman Collection Variables, and start making requests. If you do not know your ElevateAI API token, navigate to the [Manage Tokens Dashboard](#) to retrieve it.

Get started today and unlock the full potential of ElevateAI!

Download the Collection

Follow the link below to fork the ElevateAI Postman Collection into your desired Postman workspace.

[ElevateAI Postman Collection](#)

If you prefer Postman generated API documentation, then feel free to access ours at:

[ElevateAI API Postman Documentation](#)

The Postman documentation makes understanding and trying the API very easy. The main ElevateAI documentation pages should be used as the definitive reference as they include all options and parameters for every endpoint.

Using the Collection

The first step to use the API, is performing a [Declare](#) request. A Declare can be performed either with a download URI for the media to be processed or in anticipation of a subsequent [Upload](#) request. There are examples of each in the collection.

Once you have performed the [Declare](#) request, a Postman script will add the `interactionIdentifier` in the response to the Postman Collection Variable *MyInteractionId*. This will make all subsequent API requests relative to this interaction.

To perform the [Upload](#) request, you must first identify the media to be uploaded for processing. This is done under the form-data section of the request Body. Change the Key to be `filename.<extension>`, where `<extension>` is the file extension of the file you will be uploaded. Then set the Value to the file you wish to upload.

Once a file has been submitted for processing, either through the `downloadUri` method or the `upload` method, you must then wait until processing is completed before attempting to get the transcript or AI results for this interaction.

The [Status](#) endpoint is used to determine when the processing has been completed.

The only other requests which require specified input are the [Ask ElevateAI](#) endpoints. The collection includes three (3) examples for this endpoint. Feel free to adjust the questions asked in the Body for any/all of these examples.

Contact us at support@elevateai.com for further assistance.

4. Submit an Interaction

4.1. Declare an Audio Interaction

// Overview

The first step in processing any interaction is to **declare the interaction**.

When the **POST** request successfully executes, an HTTP status is returned to indicate the request was successful, along with a JSON response containing the `interactionIdentifier`.

Sample API Response



```
{
  "interactionIdentifier": "2ebaffd6-7a4b-4bcf-a169-678af0705779"
}
```

It is important you store the `interactionIdentifier` as it will serve as a required parameter for all other ElevateAI endpoints, enabling you to **POST**, **GET**, and/or **DELETE** information associated with this interaction.

If an error occurs when requesting to declare the interaction, a standard HTTP response code is returned to indicate the request was unsuccessful, along with a JSON response containing additional details to assist in [troubleshooting](#).

All interactions **successfully declared** will count towards your monthly [request quota](#), irrespective of file upload status or processing outcome.

Only interactions **successfully processed** are eligible for billing. Once declared, you can confirm the [processing status](#) of the interaction as well as track impacts to [usage](#) and [spend](#).

Echo

Introducing **Echo**, our most accurate transcription model ever!

Echo delivers a **significant leap in accuracy**, outperforming our previous CX-focused model by 40%, ensuring superior transcription quality across every use case.

To take advantage of the Echo model, add the body parameter "model" to your POST declare and set the value to "echo".

NOTE: Currently the Echo model *does not* perform redaction. Future releases will continue to move Echo towards feature parity with ElevateAI's CX transcription model. Feature updates can be seen via our [Release Notes](#).

// Request Parameters & Code Examples

Declare an Interaction for Processing (Audio)

POST

<https://api.elevateai.com/v1/interactions>



Request

HEADER PARAMETERS

X-API-Token

String

required

Valid API key associated with your account for authentication and usage tracking

Content-Type

String

required

Will be 'application/JSON'

BODY PARAMETERS

type

String

required

Type of interaction to be processed, will be "audio"

languageTag

String

optional

Specifies the language to use when processing the audio. NOTE: For the Echo model, use "auto" to automatically determine the language (over 50 supported).

For the CX model, the options are:

"en-us" (for North American English),

"en" (for International English),

"es-419" (for North American Spanish),

"pt-br" for Brazilian Portuguese or

"ja" for Japanese.

Omitting this parameter will default to "auto" for model echo and "en-us" for model cx.

vertical

String

optional

Vertical associated with interaction, will be "default"

model String optional

Defines the transcription model to be used. Available options are "cx" and "echo". Default is "cx".

audioTranscriptionMode String optional

NOTE: Effective v1.11 on February 29, 2024 all declared interactions will be processed with "highAccuracy".

This previously required parameter has been deprecated and is no longer required and any value will be ignored.

originalFileName String optional

The filename of the original file for this interaction

externalIdentifier String optional

Any external identifier that the user would like associated with this interaction

downloadUri String optional

The URL of your audio file

Important: Do not include the downloadUri parameter if you wish to upload a local audio file; an empty string will cause the download to fail

downloadAuth ▶ Object optional

This optional body parameter can be used when the downloadUri is specified and the download source requires authentication. NOTE: If an expiring token is used for authentication, make sure it has a reasonable amount of time left prior to expiration.

includeAiResults Boolean optional

NOTE: Effective v1.11 on February 29, 2024 all declared interactions will be processed with includeAiResults set to 'true'.

metadata

String

optional

JSON payload containing the metadata for this audio file. Must only be used if the account has Explore enabled. If not, specifying this element will cause a response of 400 Bad Request.



Curl



Node.js



JS



Python



Ruby



C#

PHP



```
curl --location --request POST 'https://api.elevateai.com/v1/interactions' \
--header 'Content-Type: application/json' \
--header 'X-API-Token: {Your API Token}' \
--data-raw '{
  "type": "audio",
  "languageTag": "auto",
  "vertical": "default",
  "model": "echo",
  "audioTranscriptionMode": "highAccuracy",
  "originalFileName": "ABC00012.wav",
  "includeAiResults": true,
  "metadata": "{Your JSON metada payload}"
}'
```

Responses

● 201

● 400

● 401

● 429



```
{
  "string"
}
```

Details on transcription and CX AI features supported by language can be found on the [overview page](#).

// Submitting an Audio File for Processing

Download URI

Providing access to audio files via the `downloadUri` parameter is highly recommended for fastest [processing times](#).

When using the Download URI option, authentication parameters can be included in the Declare for source systems which require it. See Body Parameters above.

Local File

If the `downloadUri` parameter is not included when declaring the interaction, the audio should be [uploaded to the API directly](#) for processing.

// Submitting metadata

Metadata must only be submitted with an API-Token for an account which has Explore enabled. Use under other circumstances will cause the Declare to fail, and a response of 400 Bad Request.

For details on the types and formatting of metadata, please see [Metadata](#).

4.1.1. Metadata

// Metadata and Explore

The use of the metadata parameter when declaring an interaction allows for the importation of additional information about the interaction which will be useful when performing analysis and reporting with ElevateAI Explore.

If the metadata body parameter is specified in a Declare for an account that is not enabled for Explore, the operation will return a response of **400 Bad Request**.

// Metadata Format

ElevateAI has both standard metadata fields and custom fields. The standard fields are those most commonly found in contact center call recording environments and provide key information for identifying the details of an interaction.

Though the values of all fields will be strings when passed into the Declare operation, the values for certain fields must fit the expected type of that field.

DateTime fields must be in the ISO 8601 (UTC) format shown in the JSON example below, with the T time introduction.

The custom fields all fall within the "custom" section. The values within these entries will all be required to match the associated type: dateTime, text, integer and decimal.

The number of available fields for the custom field types is as follows:

Field Type	Count
dateTime	45
text	165
integer	50
decimal	40

Custom field names are in the format `custom.{types}.{type}#`. For example the 8th integer field is 'custom.integers.integer8'.

Any fields which do not map to an ElevateAI field are ignored without feedback.

Following are a metadata sample in JSON format and a class specification for the complete metadata schema.

JSON

 C#



```
namespace ElevateAI
{
    internal class EaiMetadata
    {
        public string? Category { get; set; }
        public string? Direction { get; set; }
        public string? Recorded { get; set; }
        public EaiMetaAudio? Audio { get; set; }
        public EaiMetaAgent? Agent { get; set; }
        public EaiMetaSite? Site { get; set; }
        public EaiMetaCustomer? Customer { get; set; }
        public EaiMetaCustom? Custom { get; set; }
    }

    public class EaiMetaAudio
    {
        public string? Extension { get; set; }
        public string? DNIS { get; set; }
    }

    public class EaiMetaAgent
    {
        public string? Id { get; set; }
        public string? Name { get; set; }
        public EaiMetaSupervisor? Supervisor { get; set; }
        public EaiMetaGroup? Group { get; set; }
    }

    public class EaiMetaSupervisor
    {
        public string? Id { get; set; }
        public string? Name { get; set; }
    }

    public class EaiMetaGroup
    {
        public string? Id { get; set; }
        public string? Name { get; set; }
    }

    public class EaiMetaSite
    {
        public string? Id { get; set; }
        public string? Name { get; set; }
    }
}
```

```
public class EaiMetaCustomer
{
    public string? Id { get; set; }
    public string? Name { get; set; }
    public string? City { get; set; }
    public string? State { get; set; }
}

public class EaiMetaCustom
{
    public EaiDateTime? DateTimes { get; set; }
    public EaiText? Texts { get; set; }
    public EaiInteger? Integers { get; set; }
    public EaiDecimal? Decimals { get; set; }
}

public class EaiDateTime
{
    public DateTime? DateTime1 { get; set; }
    public DateTime? DateTime2 { get; set; }
    public DateTime? DateTime3 { get; set; }
    public DateTime? DateTime4 { get; set; }
    public DateTime? DateTime5 { get; set; }
    public DateTime? DateTime6 { get; set; }
    public DateTime? DateTime7 { get; set; }
    public DateTime? DateTime8 { get; set; }
    public DateTime? DateTime9 { get; set; }
    public DateTime? DateTime10 { get; set; }
    public DateTime? DateTime11 { get; set; }
    public DateTime? DateTime12 { get; set; }
    public DateTime? DateTime13 { get; set; }
    public DateTime? DateTime14 { get; set; }
    public DateTime? DateTime15 { get; set; }
    public DateTime? DateTime16 { get; set; }
    public DateTime? DateTime17 { get; set; }
    public DateTime? DateTime18 { get; set; }
    public DateTime? DateTime19 { get; set; }
    public DateTime? DateTime20 { get; set; }
    public DateTime? DateTime21 { get; set; }
    public DateTime? DateTime22 { get; set; }
    public DateTime? DateTime23 { get; set; }
    public DateTime? DateTime24 { get; set; }
    public DateTime? DateTime25 { get; set; }
    public DateTime? DateTime26 { get; set; }
    public DateTime? DateTime27 { get; set; }
```

```
public DateTime? DateTime28 { get; set; }
public DateTime? DateTime29 { get; set; }
public DateTime? DateTime30 { get; set; }
public DateTime? DateTime31 { get; set; }
public DateTime? DateTime32 { get; set; }
public DateTime? DateTime33 { get; set; }
public DateTime? DateTime34 { get; set; }
public DateTime? DateTime35 { get; set; }
public DateTime? DateTime36 { get; set; }
public DateTime? DateTime37 { get; set; }
public DateTime? DateTime38 { get; set; }
public DateTime? DateTime39 { get; set; }
public DateTime? DateTime40 { get; set; }
public DateTime? DateTime41 { get; set; }
public DateTime? DateTime42 { get; set; }
public DateTime? DateTime43 { get; set; }
public DateTime? DateTime44 { get; set; }
public DateTime? DateTime45 { get; set; }
}
```

```
public class EaiText
{
    public string? Text1 { get; set; }
    public string? Text2 { get; set; }
    public string? Text3 { get; set; }
    public string? Text4 { get; set; }
    public string? Text5 { get; set; }
    public string? Text6 { get; set; }
    public string? Text7 { get; set; }
    public string? Text8 { get; set; }
    public string? Text9 { get; set; }
    public string? Text10 { get; set; }
    public string? Text11 { get; set; }
    public string? Text12 { get; set; }
    public string? Text13 { get; set; }
    public string? Text14 { get; set; }
    public string? Text15 { get; set; }
    public string? Text16 { get; set; }
    public string? Text17 { get; set; }
    public string? Text18 { get; set; }
    public string? Text19 { get; set; }
    public string? Text20 { get; set; }
    public string? Text21 { get; set; }
    public string? Text22 { get; set; }
    public string? Text23 { get; set; }
    public string? Text24 { get; set; }
}
```



```
public string? Text25 { get; set; }
public string? Text26 { get; set; }
public string? Text27 { get; set; }
public string? Text28 { get; set; }
public string? Text29 { get; set; }
public string? Text30 { get; set; }
public string? Text31 { get; set; }
public string? Text32 { get; set; }
public string? Text33 { get; set; }
public string? Text34 { get; set; }
public string? Text35 { get; set; }
public string? Text36 { get; set; }
public string? Text37 { get; set; }
public string? Text38 { get; set; }
public string? Text39 { get; set; }
public string? Text40 { get; set; }
public string? Text41 { get; set; }
public string? Text42 { get; set; }
public string? Text43 { get; set; }
public string? Text44 { get; set; }
public string? Text45 { get; set; }
public string? Text46 { get; set; }
public string? Text47 { get; set; }
public string? Text48 { get; set; }
public string? Text49 { get; set; }
public string? Text50 { get; set; }
public string? Text51 { get; set; }
public string? Text52 { get; set; }
public string? Text53 { get; set; }
public string? Text54 { get; set; }
public string? Text55 { get; set; }
public string? Text56 { get; set; }
public string? Text57 { get; set; }
public string? Text58 { get; set; }
public string? Text59 { get; set; }
public string? Text60 { get; set; }
public string? Text61 { get; set; }
public string? Text62 { get; set; }
public string? Text63 { get; set; }
public string? Text64 { get; set; }
public string? Text65 { get; set; }
public string? Text66 { get; set; }
public string? Text67 { get; set; }
public string? Text68 { get; set; }
public string? Text69 { get; set; }
public string? Text70 { get; set; }
```

```
public string? Text71 { get; set; }
public string? Text72 { get; set; }
public string? Text73 { get; set; }
public string? Text74 { get; set; }
public string? Text75 { get; set; }
public string? Text76 { get; set; }
public string? Text77 { get; set; }
public string? Text78 { get; set; }
public string? Text79 { get; set; }
public string? Text80 { get; set; }
public string? Text81 { get; set; }
public string? Text82 { get; set; }
public string? Text83 { get; set; }
public string? Text84 { get; set; }
public string? Text85 { get; set; }
public string? Text86 { get; set; }
public string? Text87 { get; set; }
public string? Text88 { get; set; }
public string? Text89 { get; set; }
public string? Text90 { get; set; }
public string? Text91 { get; set; }
public string? Text92 { get; set; }
public string? Text93 { get; set; }
public string? Text94 { get; set; }
public string? Text95 { get; set; }
public string? Text96 { get; set; }
public string? Text97 { get; set; }
public string? Text98 { get; set; }
public string? Text99 { get; set; }
public string? Text100 { get; set; }
public string? Text101 { get; set; }
public string? Text102 { get; set; }
public string? Text103 { get; set; }
public string? Text104 { get; set; }
public string? Text105 { get; set; }
public string? Text106 { get; set; }
public string? Text107 { get; set; }
public string? Text108 { get; set; }
public string? Text109 { get; set; }
public string? Text110 { get; set; }
public string? Text111 { get; set; }
public string? Text112 { get; set; }
public string? Text113 { get; set; }
public string? Text114 { get; set; }
public string? Text115 { get; set; }
public string? Text116 { get; set; }
```

```
public string? Text117 { get; set; }
public string? Text118 { get; set; }
public string? Text119 { get; set; }
public string? Text120 { get; set; }
public string? Text121 { get; set; }
public string? Text122 { get; set; }
public string? Text123 { get; set; }
public string? Text124 { get; set; }
public string? Text125 { get; set; }
public string? Text126 { get; set; }
public string? Text127 { get; set; }
public string? Text128 { get; set; }
public string? Text129 { get; set; }
public string? Text130 { get; set; }
public string? Text131 { get; set; }
public string? Text132 { get; set; }
public string? Text133 { get; set; }
public string? Text134 { get; set; }
public string? Text135 { get; set; }
public string? Text136 { get; set; }
public string? Text137 { get; set; }
public string? Text138 { get; set; }
public string? Text139 { get; set; }
public string? Text140 { get; set; }
public string? Text141 { get; set; }
public string? Text142 { get; set; }
public string? Text143 { get; set; }
public string? Text144 { get; set; }
public string? Text145 { get; set; }
public string? Text146 { get; set; }
public string? Text147 { get; set; }
public string? Text148 { get; set; }
public string? Text149 { get; set; }
public string? Text150 { get; set; }
public string? Text151 { get; set; }
public string? Text152 { get; set; }
public string? Text153 { get; set; }
public string? Text154 { get; set; }
public string? Text155 { get; set; }
public string? Text156 { get; set; }
public string? Text157 { get; set; }
public string? Text158 { get; set; }
public string? Text159 { get; set; }
public string? Text160 { get; set; }
public string? Text161 { get; set; }
public string? Text162 { get; set; }
```

```
    public string? Text163 { get; set; }
    public string? Text164 { get; set; }
    public string? Text165 { get; set; }
}
```

```
public class EaiInteger
{
    public int? Integer1 { get; set; }
    public int? Integer2 { get; set; }
    public int? Integer3 { get; set; }
    public int? Integer4 { get; set; }
    public int? Integer5 { get; set; }
    public int? Integer6 { get; set; }
    public int? Integer7 { get; set; }
    public int? Integer8 { get; set; }
    public int? Integer9 { get; set; }
    public int? Integer10 { get; set; }
    public int? Integer11 { get; set; }
    public int? Integer12 { get; set; }
    public int? Integer13 { get; set; }
    public int? Integer14 { get; set; }
    public int? Integer15 { get; set; }
    public int? Integer16 { get; set; }
    public int? Integer17 { get; set; }
    public int? Integer18 { get; set; }
    public int? Integer19 { get; set; }
    public int? Integer20 { get; set; }
    public int? Integer21 { get; set; }
    public int? Integer22 { get; set; }
    public int? Integer23 { get; set; }
    public int? Integer24 { get; set; }
    public int? Integer25 { get; set; }
    public int? Integer26 { get; set; }
    public int? Integer27 { get; set; }
    public int? Integer28 { get; set; }
    public int? Integer29 { get; set; }
    public int? Integer30 { get; set; }
    public int? Integer31 { get; set; }
    public int? Integer32 { get; set; }
    public int? Integer33 { get; set; }
    public int? Integer34 { get; set; }
    public int? Integer35 { get; set; }
    public int? Integer36 { get; set; }
    public int? Integer37 { get; set; }
    public int? Integer38 { get; set; }
    public int? Integer39 { get; set; }
}
```

```
    public int? Integer40 { get; set; }
    public int? Integer41 { get; set; }
    public int? Integer42 { get; set; }
    public int? Integer43 { get; set; }
    public int? Integer44 { get; set; }
    public int? Integer45 { get; set; }
    public int? Integer46 { get; set; }
    public int? Integer47 { get; set; }
    public int? Integer48 { get; set; }
    public int? Integer49 { get; set; }
    public int? Integer50 { get; set; }
}
```

```
public class EaiDecimal
{
    public decimal? Decimal1 { get; set; }
    public decimal? Decimal2 { get; set; }
    public decimal? Decimal3 { get; set; }
    public decimal? Decimal4 { get; set; }
    public decimal? Decimal5 { get; set; }
    public decimal? Decimal6 { get; set; }
    public decimal? Decimal7 { get; set; }
    public decimal? Decimal8 { get; set; }
    public decimal? Decimal9 { get; set; }
    public decimal? Decimal10 { get; set; }
    public decimal? Decimal11 { get; set; }
    public decimal? Decimal12 { get; set; }
    public decimal? Decimal13 { get; set; }
    public decimal? Decimal14 { get; set; }
    public decimal? Decimal15 { get; set; }
    public decimal? Decimal16 { get; set; }
    public decimal? Decimal17 { get; set; }
    public decimal? Decimal18 { get; set; }
    public decimal? Decimal19 { get; set; }
    public decimal? Decimal20 { get; set; }
    public decimal? Decimal21 { get; set; }
    public decimal? Decimal22 { get; set; }
    public decimal? Decimal23 { get; set; }
    public decimal? Decimal24 { get; set; }
    public decimal? Decimal25 { get; set; }
    public decimal? Decimal26 { get; set; }
    public decimal? Decimal27 { get; set; }
    public decimal? Decimal28 { get; set; }
    public decimal? Decimal29 { get; set; }
    public decimal? Decimal30 { get; set; }
    public decimal? Decimal31 { get; set; }
}
```

```
public decimal? Decimal32 { get; set; }  
public decimal? Decimal33 { get; set; }  
public decimal? Decimal34 { get; set; }  
public decimal? Decimal35 { get; set; }  
public decimal? Decimal36 { get; set; }  
public decimal? Decimal37 { get; set; }  
public decimal? Decimal38 { get; set; }
```

4.1.2. Upload an Audio File

// Overview

Leveraging the unique `interactionIdentifier` provided upon successfully declaring the `audio` interaction, this endpoint enables you to **upload** a valid audio for processing directly from the host machine as a `multipart/form-data` request.

Pro tip: Providing access to audio files via the `downloadUri` is highly recommended for fastest processing times.

When this `POST` request successfully executes, an HTTP status is returned to indicate the request was successful and the file is in the process of being uploaded by the Service.

If an error occurs when requesting to upload interaction, a standard HTTP response code is returned to indicate the request was unsuccessful, along with a JSON response containing additional details to assist in troubleshooting.

Remember: You can easily retrieve the processing status of your interaction at any time!

// Request Parameters & Code Examples

Upload an Interaction (Audio)

POST

<https://api.elevateai.com/v1/interactions/{interactionIdentifier}> 

Request

HEADER PARAMETERS

X-API-Token

String

required

Valid API key associated with your account for authentication and usage tracking

FORM PARAMETERS

filename.{extension}

String

required

Parameter name will include the file type extension (example: 'filename.wav') with key providing path to local file



Curl



Node.js



JS



Python



Ruby



C#

PHP



```
var request = require('request');
var fs = require('fs');
var path = require('path');

var filepath = '/path/to/file'

var options = {
  'method': 'POST',
  'url': 'https://api.elevateai.com/v1/interactions/{interactionIdentifier}',
  'headers': {
    'X-API-Token': '{Your-API-Token}'
  },
  formData: {
    'filename.wav': {
      'value': fs.createReadStream(filepath),
      'options': {
        'filename': path.parse(filepath).base,
        'contentType': null
      }
    }
  }
};

request(options, function (error, response) {
  if (error) throw new Error(error);
  console.log(response.body);
});
```

Responses

● 201● 400● 401● 409● 413

4.2. Declare a Transcript

// Overview

The first step in processing any interaction is to **declare the interaction**.

When the **POST** request successfully executes, an HTTP status is returned to indicate the request was successful, along with a JSON response containing the `interactionIdentifier`.

Sample API Response



```
{
  "interactionIdentifier": "46985457-fa6b-4071-a3f6-5c4858160ec9"
}
```

It is important you store the `interactionIdentifier` as it will serve as a required parameter for all other ElevateAI endpoints, enabling you to **POST**, **GET**, and/or **DELETE** information associated with this interaction.

If an error occurs when requesting to declare the interaction, a standard HTTP response code is returned to indicate the request was unsuccessful, along with a JSON response containing additional details to assist in [troubleshooting](#).

All interactions **successfully declared** will count towards your monthly [request quota](#), irrespective of file upload status or processing outcome.

Only interactions **successfully processed** are eligible for billing. Once declared, you are able to confirm the [processing status](#) of the interaction as well as track impacts to [usage](#) and [spend](#).

// Request Parameters & Code Examples

Details on CX AI features supported by language can be found on the [overview page](#).

Declare an Interaction for Processing (Transcript)

POST

https://api.elevateai.com/v1/interactions



Request

HEADER PARAMETERS

X-API-Token

String

required

Valid API key associated with your account for authentication and usage tracking

Content-Type

String

required

Will be 'application/JSON'

BODY PARAMETERS

type

String

required

Type of interaction to be processed, will be "transcript"

languageTag

String

required

Primary language of transcript, will be "en-us" (for North American English), "en" (for International English), or "es-419" (for North American Spanish)

vertical

String

required

Vertical associated with interaction, will be "default"

transcript ▶

Object

required

transcript text to be analyzed (see below)



Curl



Node.js



JS



Python



```
curl --location --request POST 'https://api.elevateai.com/v1/interactions' \
--header 'Content-Type: application/json' \
--header 'X-API-Token: {Your API Token}' \
--data-raw '{
  "type": "transcript",
  "languageTag": "en-us",
  "vertical": "default",
  "transcript": {
    "allParticipants": {
      "phrases": [
        "{word A}",
        "{word B}"
      ]
    }
  }
}
```

Responses

● 201 ● 400 ● 401 ● 429



```
{
  "interactionIdentifier": "string"
}
```

Search and replace text in the transcript.

// Formatting a Transcript for Processing

Transcript files will be formatted according to the schema below, and included as a body parameter when declaring the transcript for processing.

✓ Sample Transcription Format

Sample Transcription Format



```
{
  "allParticipants": {
    "phrases": [
      "I", "Hope", "This", "Documentation", "Was", "Useful", "Thank", "
      ...
    ]
    "phraseSegments": [
      {
        "startTimeOffset": 870,
        "endTimeOffset": 1330,
        "phraseIndex" 0,
        "score": 1
      }
      ...
    ]
  },
  "participantOne": {
    "phrases": [
      "I","Hope","This","Documentation","Was","Useful",
      ...
    ]
    ...
  },
  ...
}
```

Schema

Element	Type	Description
{participant}	object	Top level for speaker label , will be "allParticipants" , or "participantOne" , or "participantTwo"
{participant}/phrases	array of strings	Ordered list of each unique phase spoken by {participant} in the interaction
{participant}/phraseSegments	array of objects	List of details associated with each phrase in transcript for the {participant}
{participant}/phraseSegments/startTimeOffset	number	Start time of phrase (in milliseconds)
{participant}/phraseSegments/endTimeOffset	number	End time of phrase (in milliseconds)
{participant}/phraseSegments/phraseIndex	number	Index of phrase in {participant}/phrases
{participant}/phraseSegments/score	float	Confidence score, if not available should be 1

Pro tip: `allParticipants` is a required object when providing a transcript for analysis. While `participantOne` and `participantTwo` objects are not required, they are highly recommended.

4.3. Declare a Chat Interaction

// Overview

The first step in processing any interaction is to **declare the interaction**.

When the **POST** request successfully executes, an HTTP status is returned to indicate the request was successful, along with a JSON response containing the `interactionIdentifier`.

Sample API Response



```
{
  "interactionIdentifier": "6091d697-9a98-4953-9850-d3e9b82cd5ab"
}
```

It is important you store the `interactionIdentifier` as it will serve as a required parameter for all other ElevateAI endpoints, enabling you to **POST**, **GET**, and/or **DELETE** information associated with this interaction.

If an error occurs when requesting to declare the interaction, a standard HTTP response code is returned to indicate the request was unsuccessful, along with a JSON response containing additional details to assist in [troubleshooting](#).

All interactions **successfully declared** will count towards your monthly [request quota](#), irrespective of file upload status or processing outcome.

Only interactions **successfully processed** are eligible for billing. Once declared, you are able to confirm the [processing status](#) of the interaction as well as track impacts to [usage](#) and [spend](#).

// Request Parameters & Code Samples

Details on CX AI features supported by language can be found on the [overview page](#).

Declare a Chat Interaction

POST

https://api.elevateai.com/v1/interactions



Request

HEADER PARAMETERS

X-API-Token

String

required

Valid API key associated with your account for authentication and usage tracking

Content-Type

String

required

Will be 'application/JSON'

BODY PARAMETERS

type

String

required

Type of interaction to be processed, will be "chat"

languageTag

String

required

Primary language of the chat, will be "en-us" (for North American English), "en" (for International English), or "es-419" (for North American Spanish)

vertical

String

required

Vertical associated with interaction, will be "default"

chat ▶

Object

required

Formatted chat text (see below)

 Curl Node.js JS JS Python Ruby





```
var myHeaders = new Headers();
myHeaders.append("X-API-Token", "string");
myHeaders.append("Content-Type", "string");

var raw = "{\"type\":\"String\",\"languageTag\":\"String\",\"vertical\":\""

var requestOptions = {
  method: 'POST',
  headers: myHeaders,
  body: raw,
  redirect: 'follow'
};

fetch("https://api.elevateai.com/v1/interactions", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

Responses

 201  400  401  429



```
{
  "errorMessage": "string"
}
```

// Formatting a Chat for Processing

Chat files will be formatted according to the schema below, and included as a body parameter when declaring a chat interaction for processing.

✓ Sample Chat File Format

Sample Chat Format



```
{
  "participants": [
    {
      "participant_id": "ElevateAI Team",
      "name": "Madison",
      "role": "agent"
    },
    {
      "participant_id": "ElevateAI Evangelist",
      "name": "Developer",
      "role": "customer"
    }
  ],
  "messages": [
    {
      "participant_id": "ElevateAI Team",
      "timestamp": "2022-07-04T14:25:15Z",
      "content": "We hope this documentation is useful!"
    },
    {
      "participant_id": "ElevateAI Evangelist",
      "timestamp": "2022-07-04T14:26:45Z",
      "content": "It is, thank you!"
    }
  ]
}
```

▼ Schema

Element	Type	Description
participants	array of objects	Ordered list of details associated with each participant, will include 1+ objects
participant/ participant_id	string	Unique identifier associated with participant
participants/ name	string	Name of participant
participants/ role	string or null	Will be <code>agent</code> or <code>customer</code>
messages	array of objects	Ordered list of details associated with each chat message, will include 1+ objects
messages/ participant_id	string	Maps to a unique participant_id within participants object
messages/ timestamp	string	Date/time that the chat message was sent, formatted according to ISO 8601 (UTC)
messages/ content	string	Chat message

4.4. Untitled

5. Check the Status

5.1. Check the Processing Status

// Overview

Leveraging the unique `interactionIdentifier` returned upon successfully declaring the [audio](#), [transcript](#), or [chat](#) interaction, this endpoint enables you to instantly retrieve detailed updates on the **current status** of your interaction analysis.

Pro tip: We recommend checking the status **at most** every 30 seconds for an audio file and every 5 seconds for a chat or transcript.

When this `GET` request successfully executes, an HTTP status is returned to indicate the request was successful, along with a JSON response providing [details](#) on the processing status of your interaction.

If an error occurs when requesting the status, a standard HTTP response code is returned to indicate the request was unsuccessful, along with a JSON response containing additional details to assist in [troubleshooting](#).

Sample API Response

```
{
  "identifier": "bd5c92f3-c228-44a6-
  "status": "processed",
  "declared": "2022-06-17T21:23:06.1
  "uploaded": null,
  "downloaded": "2022-06-17T21:24:53
  "processed": "2022-06-17T21:29:05.
  "errorMessage": null
}
```


You are only able to retrieve the [phrase-by-phrase transcript](#), [punctuated transcript](#) and/or [CX AI](#) results associated with the interaction once the `status` indicates the interaction has been `processed`.

Note: Only interactions that have been `processed` will result in applicable usage billing. See the ElevateAI [Invoicing Policy](#) for more information.

// Request Parameters & Code Example

Retrieve Processing Status

GET

https://api.elevateai.com/v1/interactions/{interactionIdentifier} 

Request

PATH PARAMS

interactionIdentifier String **required**

interactionIdentifier returned upon successfully declaring the interaction

HEADER PARAMETERS

X-API-Token String **required**

Valid API key associated with your account for authentication and usage tracking

Accept-Encoding String **optional**

Recommended for bandwidth optimization, will be 'gzip, deflate, br'



Curl



Node.js



JS



Python



Ruby



C#

PHP



```
import http.client

conn = http.client.HTTPSConnection("api.elevateai.com")
payload = ''
headers = {
    'Accept-Encoding': 'gzip, deflate, br',
    'X-API-Token': '{Your API Token}'
}
conn.request("GET", "/v1/interactions/{InteractionIdentifier}/status", payload)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

Responses

● 200● 401● 404

```
{
  "identifier": "string",
  "status": "string",
  "declared": "string",
  "uploaded": "string",
  "downloaded": "string",
  "processed": "string",
  "errorMessage": "string"
}
```

// Response Schema

Schema

Element	Type	Description
identifier	string	Unique <code>interactionIdentifier</code> associated with the request
status	string	Current status of interaction
declared	string	Date/Time, formatted according to ISO 8601 (UTC) , that the interaction was successfully declared
uploaded	string or null	If applicable, Date/Time formatted according to ISO 8601 (UTC) that an interaction provided directly to the API was successfully uploaded to ElevateAI and ready to be entered into the processing queue
downloaded	string or null	If applicable, Date/Time formatted according to ISO 8601 (UTC) that an interaction provided via <code>downloadUri</code> was successfully downloaded and ready to be entered into the processing queue
processed	string or null	If applicable, Date/Time formatted according to ISO 8601 (UTC) that processing was complete and results available to be retrieved
errorMessage	string or null	If applicable, human-readable description of errors encountered when uploading, downloading, or processing the interaction that must be addressed prior to re-declaring an interaction. If you need additional support, please contact us at support@elevateai.com and include the message provided (Note: Typically, declaring the same call again will not resolve the issue).

Note: All source data has been deleted once `status` indicates either `processed` or `processingFailed`.

Status Definitions

Status	Description
declared	An <code>audio</code> interaction has been successfully declared with a <code>downloadUri</code> and the file is waiting to be inserted into the download queue OR a <code>transcript</code> or <code>chat</code> interaction has been successfully declared and is waiting to be inserted in the processing queue
filePendingUpload	An <code>audio</code> interaction has been successfully declared without a <code>downloadUri</code> , and is waiting for the associated file to be uploaded directly to the API
fileUploading	An <code>audio</code> file associated with this interaction has been provided directly to the API and is in the process of being uploaded by the Service
fileUploaded	An <code>audio</code> file associated with this interaction has been successfully uploaded and is waiting to be inserted into the processing queue
fileUploadFailed	There was an error encountered when uploading an <code>audio</code> file to the ElevateAI system - reference additional information returned in the <code>errorMessage</code> element re-upload the file for processing
filePendingDownload	An <code>audio</code> interaction has been successfully declared with a <code>downloadUri</code> , and the file is in queue to be downloaded
fileDownloading	An <code>audio</code> interaction has been successfully declared with a <code>downloadUri</code> and the file is in the process of being download from the URL provided
fileDownloaded	An <code>audio</code> interaction has been successfully downloaded from the provided <code>downloadUri</code> URL, and is waiting to be inserted into the processing queue
fileDownloadFailed	There was an error encountered when downloading an <code>audio</code> file to the ElevateAI system from the <code>downloadUri</code> URL - reference additional information returned in the <code>errorMessage</code> element and confirm you have provided a valid file format prior to re-declaring
pendingProcessing	The <code>audio</code> , <code>transcript</code> , or <code>chat</code> interaction is in queue for processing

Status	Description
processing	The <code>audio</code> , <code>transcript</code> , or <code>chat</code> interaction is actively being processed by the Service
processed	The <code>audio</code> , <code>transcript</code> , or <code>chat</code> interaction has been successfully processed and you are now able to retrieve the phrase-by-phrase transcript , punctuated transcript and/or CX AI results associated with the interaction
processingFailed	There was an error encountered while processing the <code>audio</code> , <code>transcript</code> or <code>chat</code> interaction – reference additional information returned in the <code>errorMessage</code> element and contact us at support@elevateai.com with any additional questions (Note: Typically, declaring the same call again will not resolve the issue).

5.2. Retrieve Declared Parameters

// Overview

Leveraging the unique `interactionIdentifier` provided upon successfully declaring the `audio`, `transcript`, or `chat` interaction, this endpoint enables you to instantly **retrieve the parameters declared** when submitting the interaction for processing.

When this `GET` request successfully executes, an HTTP status will be returned to indicate the request was successful, along with a JSON response outlining the [declared parameters](#) for the given `interactionIdentifier`.

Sample API Response



```
{
  "identifier": "a3c1c9b5-5371-472c-97c1-538b369ff149",
  "type": "audio",
  "languageTag": "en-US",
  "vertical": "default",
  "audioTranscriptionMode": "highSpeed",
  "transcript": null,
  "chat": null,
  "downloadUri": null,
  "includeAiResults": true
}
```

If an error occurs when requesting to declare the interaction, a standard HTTP response code is returned to indicate the request was unsuccessful, along with a JSON response containing additional details to assist in [troubleshooting](#).

// Request Parameters & Code Example

Retrieve Declared Parameters

GET

https://api.elevateai.com/v1/interactions/{interactionIdentifier} 

Request

PATH PARAMS

interactionIdentifier String **required**

interactionIdentifier returned upon successfully declaring the interaction

HEADER PARAMETERS

X-API-Token String **required**

Valid API key associated with your account for authentication and usage tracking

Accept-Encoding String **optional**

Recommended for bandwidth optimization, will be 'gzip, deflate, br'



Curl



Node.js



JS



Python



Ruby



C#

PHP



```
import http.client

conn = http.client.HTTPSConnection("api.elevateai.com")
payload = ''
headers = {
    'Accept-Encoding': 'gzip, deflate, br',
    'X-API-Token': '{Your API Token}'
}
conn.request("GET", "/v1/interactions/{InteractionIdentifier}", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

Responses

200

401

404



```
{
  "identifier": "string",
  "type": "string",
  "languageTag": "string",
  "vertical": "string",
  "audioTranscriptionMode": "string",
  "transcript": null,
  "chat": null,
  "downloadUri": "string",
  "includeAiResults": boolean
}
```

// Response Schema

Schema

Element	Type	Description
identifier	string	Unique <code>interactionIdentifier</code> associated with the request
type	string	Type of interaction, will be "audio", "transcript", or "chat"
languageTag	string	Primary language of the file, will be "en-us" (for North American English), "en" (for International English), or "es-419" (for North American Spanish)
vertical	string	Vertical associated with interaction, will be "default"
audioTranscriptionMode	string or null	If applicable, desired transcription prioritization, will be "highSpeed" or "highAccuracy"
transcript	null	<i>Uploaded transcripts are deleted upon processing</i>
chat	null	<i>Uploaded chats are deleted upon processing</i>
downloadUri	string or null	URL of audio file, if applicable
includeAiResults	string	Include AI results, will be true or false

6. Retrieve Results

6.1. Get Phrase-by-Phrase Transcript

// Overview

Leveraging the unique `interactionIdentifier` provided upon successfully [declaring](#) the `audio` interaction, this endpoint enables you to instantly retrieve a detailed **phrase-by-phrase transcript** for the audio file provided including precise **word timings**, **confidence scores**, and [speaker labels](#).

When this `GET` successfully executes, an HTTP status is returned to indicate the request was successful, along with a JSON response providing the [detailed transcription output](#).

```
{
  "allParticipants": {
    "phrases": [
      "Thank", "You", "For", "Calling", ...
    ],
    "phraseSegments": [
      {
        "startTimeOffset": 1410,
        "endTimeOffset": 1730,
        "phraseIndex": 0,
        "score": 1
      },
      {
        "startTimeOffset": 1730,
        "endTimeOffset": 1810,
        "phraseIndex": 1,
        "score": 1
      },
      ...
    ]
  }
  ...
}
```

If an error occurs when requesting to declare the interaction, a standard HTTP status will be returned to indicate the request was unsuccessful, along with a JSON response containing additional details to assist in [troubleshooting](#).

// Request Parameters & Code Examples

Retrieve a Phrase-by-Phrase Transcript

GET

<https://api.elevateai.com/v1/interactions/{interactionIdentifier}> 

Request

PATH PARAMS

interactionIdentifier String **required**

interactionIdentifier returned upon successfully declaring the interaction

HEADER PARAMETERS

X-API-Token String **required**

Valid API key associated with your account for authentication and usage tracking

Accept-Encoding String **optional**

Recommended for bandwidth optimization, will be 'gzip, deflate, br'



Curl



Node.js



JS



Python



Ruby



C#

PHP



```
import http.client
import json

conn = http.client.HTTPSConnection("api.elevateai.com")
payload = ''
headers = {
    'X-API-Token': '{Your API Token}',
    'Content-Type': 'application/json',
    'Accept-Encoding': 'gzip, deflate, br'
}
conn.request("GET", "/v1/interactions/{interactionIdentifier}/transcript",
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

Responses

● 200

● 401

● 404



// Response Schema

Schema

Element	Type	Description
{participant}	object	Top level for speaker label , identifying whether data in object represents speech associated with "allParticipants" , "participantOne" , or "participantTwo"
{participant}/phrases	array of strings	Ordered list of each unique, non-redacted phrase spoken by {participant} in the interaction
{participant}/phraseSegments	array of objects	List of details associated with each utterance
{participant}/phraseSegments/startTimeOffset	number	Start time of phrase (in milliseconds)
{participant}/phraseSegments/endTimeOffset	number	End time of phrase (in milliseconds)
{participant}/phraseSegments/phraselIndex	number	Index of phrase in {participant}/phrases
{participant}/phraseSegments/score	float	Confidence score
redactionSegments	array of objects	List of details associated with each redacted phrase
redactionSegments/startTimeOffset	number	Start time of redacted phrase (in milliseconds)
redactionSegments/endTimeOffset	number	End time of redacted phrase (in milliseconds)
redactionSegments/result	string	Reason for redaction, will be "CVV", "Credit Card", or "Social Security"
redactionSegments/score	float	Confidence score

6.2. Get Punctuated Transcript

// Overview

Leveraging the unique `interactionIdentifier` provided upon successfully [declaring](#) the `audio` interaction, this endpoint enables you to instantly retrieve a detailed **sentence-by-sentence transcript** with precise **timings, confidence scores**, and [speaker labels](#).

When this `GET` request successfully executes, an HTTP status will be returned to indicate the request was successful, along with a JSON response providing [detailed transcription output](#).

```
{
  "sentenceSegments": [
    {
      "participant": "participantOne",
      "startTimeOffset": 1410,
      "endTimeOffset": 3210,
      "phrase": "Thank you for calling America phone service.",
      "score": 0.99137425
    },
    {
      "participant": "participantOne",
      "startTimeOffset": 3210,
      "endTimeOffset": 5570,
      "phrase": "This is Adam speaking, how may I be of service.",
      "score": 0.99325925
    },
    ...
  ],
  "redactionSegments": [
    {
      "startTimeOffset": 24650,
      "endTimeOffset": 27530,
      "result": "CVV",
      "score": 0
    },
    ...
  ]
}
```

If an error occurs when requesting to declare the interaction, a standard HTTP status will be returned to indicate the request was unsuccessful, along with a JSON response containing additional details to assist in [troubleshooting](#).

// Request Parameters & Code Examples

Retrieve a Punctuated Transcript

Get a transcript by its ID

GET

https://api.elevateai.com/v1/interactions/{interactionIdentifier} 

Request

PATH PARAMS

interactionIdentifier String **required**

interactionIdentifier (GUID) returned upon successfully declaring the interaction

HEADER PARAMETERS

X-API-Token String **required**

Valid API key associated with your account for authentication and usage tracking

Accept-Encoding String **optional**

Recommended for bandwidth optimization, will be 'gzip, deflate, br'

 Curl Node.js JS JS Python Ruby C#

PHP



```
var client = new RestClient("https://api.elevateai.com/v1/interactions/{in
client.Timeout = -1;
var request = new RestRequest(Method.GET);
request.AddHeader("X-API-Token", "{Your API Token}");
request.AddHeader("Content-Type", "application/json");
request.AddHeader("Accept-Encoding", "gzip, deflate, br");
IRestResponse response = client.Execute(request);
Console.WriteLine(response.Content);
```

Responses

 200  204  401  404



```
{
  "StatusCode": HttpStatusCode
}
```

// Response Schema

Schema

Element	Type	Description
sentenceSegments	array of objects	Ordered list of each sentence spoken, excluding redacted phrases
sentenceSegments/ participant	string	Speaker associated with sentence, will be either " participantOne " or " participantTwo "
sentenceSegments/ startTimeOffset	number	Start time of sentence (in milliseconds)
sentenceSegments/ endTimeOffset	number	End time of sentence (in milliseconds)
sentenceSegments/ phrase	string	Punctuated sentence, with any redacted segments represented by "# # # #"
sentenceSegments/ score	float	Confidence score
redactionSegments	array of objects	List of details associated with each redacted segment
redactionSegments/ startTimeOffset	number	Start time of redacted phrase (in milliseconds)
redactionSegments/ endTimeOffset	number	End time of redacted phrase (in milliseconds)
redactionSegments/ result	string	Reason for redaction, will be "CVV", "Credit Card", or "Social Security"
redactionSegments/ score	float	Confidence score

6.3. Get CX AI

// Overview

Leveraging the unique `interactionIdentifier` provided upon successfully declaring the `audio`, `transcript`, or `chat` interaction, this endpoint enables you to instantly retrieve [CX AI](#) analysis outputs.

When this `GET` request successfully executes, an HTTP status will be returned to indicate the request was successful, along with a JSON response providing [detailed CX AI output](#).



```

{
  "allParticipants": {
    "enlightenBundles": [
      {
        "name": "Customer Satisfaction",
        "models": [
          {
            "name": "AcknowledgeLoyalty",
            "score": 4.169592,
            "minScore": -3.1514606,
            "minScoreOffset": 75345,
            "maxScore": 2.7370446,
            "maxScoreOffset": 196550,
            "quartile1": -4.7154593,
            "quartile2": 0.39559406,
            "quartile3": 1.9443647,
            "quartile4": 6.545092
          },
          {
            "name": "ActiveListening",
            "score": 5.8615756,
            "minScore": -1.1147503,
            "minScoreOffset": 49390,
            "maxScore": 3.5741313,
            "maxScoreOffset": 197450,
            "quartile1": -1.8281904,
            "quartile2": 0.8814685,
            "quartile3": 1.9214101,
            "quartile4": 4.8868876
          },
          ...
        ]
      },
      ...
    ],
    "voiceActivitySegments": [
      {
        "startTimeOffset": 0,
        "endTimeOffset": 1000,
        "result": "VOICE"
      },
      {
        "startTimeOffset": 1000,
        "endTimeOffset": 39000,
        "result": "SILENCE"
      }
    ]
  }
}

```

```
    },  
    ...  
  ]  
},  
...  
}
```

If an error occurs when requesting to declare the interaction, a standard HTTP status will be returned to indicate the request was unsuccessful, along with a JSON response containing additional details to assist in [troubleshooting](#).

// Request Parameters & Code Examples

Retrieve a Punctuated Transcript

Get AI results by ID

GET

https://api.elevateai.com/v1/interactions/{interactionIdentifier} 

Request

PATH PARAMS

interactionIdentifier String required

interactionIdentifier returned upon successfully declaring the interaction

HEADER PARAMETERS

X-API-Token String required

Valid API key associated with your account for authentication and usage tracking

Accept-Encoding String optional

Recommended for bandwidth optimization, will be 'gzip, deflate, br'



Curl



Node.js



JS



Python



Ruby



C#

PHP



```
require "uri"
require "json"
require "net/http"

url = URI("https://api.elevateai.com/v1/interactions/{interactionIdentifier")

https = Net::HTTP.new(url.host, url.port)
https.use_ssl = true

request = Net::HTTP::Get.new(url)
request["X-API-Token"] = "{Your API Token}"
request["Content-Type"] = "application/json"
request["Accept-Encoding"] = "gzip, deflate, br"

response = https.request(request)
puts response.read_body
```

Responses

● 200

● 401

● 404



// Response Schema

Schema

Element	Type	Description
{participant}	object	Top level for speaker label , identifying whether output was determined based on speech across " allParticipants ", " participantOne ", or " participantTwo "
{participant}/enlightenBundles	array of objects	Top level for each eligible AI behavioral model
{participant}/enlightenBundles/name	string	Name of Enlighten Bundle associated with model, will be "Customer Satisfaction" or "Vulnerable Customer"
{participant}/enlightenBundles/models	array of objects	Ordered list of analysis outputs associated with each available model from bundle
{participant}/enlightenBundles/models/name	string	Name of model
{participant}/enlightenBundles/models/score	float	Overall score
{participant}/enlightenBundles/models/minScore	float	Lowest score calculated throughout interaction
{participant}/enlightenBundles/models/minScoreOffset	number	Time of lowest score calculated throughout interaction
{participant}/enlightenBundles/models/maxScore	float	Highest score calculated throughout interaction
{participant}/enlightenBundles/models/maxScoreOffset	number	Time of highest score calculated throughout interaction
{participant}/enlightenBundles/models/qua	float	Score calculated across first quartile of

Element	Type	Description
rtile1		interaction
{participant}/enlighte nBundles/models/ qua rtile2	float	Score calculated across second quartile of interaction
{participant}/enlighte nBundles/models/ qua rtile3	float	Score calculated across third quartile of interaction
{participant}/enlighte nBundles/models/ qua rtile4	float	Score calculated across fourth quartile of interaction
{participant}/ voiceA ctivitySegments	array of objects or null	Top level for each "voice" or "silence" segment
{participant}/voiceAc tivitySegments/ start TimeOffset	number	Start time of segment
{participant}/voiceAc tivitySegments/ endTi meOffset	number	End time of segment
{participant}/voiceAc tivitySegments/ result	string	Type of segment, will be " voice " or " silence "

// Model Scores

Below are JSON structures containing the models, score ranges, and their descriptions. Note the lower bounds are exclusive and upper bounds are inclusive.

CSAT Model Scores

 **CSATModelScores_v2.json**




```
[
  {
    "ModelName": "AcknowledgeLoyalty",
    "Ranges": [
      {
        "LowerBound": null,
        "UpperBound": -6.158,
        "IndexScore": 0,
        "ScoreDescription": "Strongly Negative"
      },
      {
        "LowerBound": -6.158,
        "UpperBound": -2.63,
        "IndexScore": 4,
        "ScoreDescription": "Moderately Negative"
      },
      {
        "LowerBound": -2.63,
        "UpperBound": 1.201,
        "IndexScore": 5,
        "ScoreDescription": "Neutral"
      },
      {
        "LowerBound": 1.201,
        "UpperBound": 7.954,
        "IndexScore": 6,
        "ScoreDescription": "Moderately Positive"
      },
      {
        "LowerBound": 7.954,
        "UpperBound": 90.437,
        "IndexScore": 10,
        "ScoreDescription": "Strongly Positive"
      }
    ]
  },
  {
    "ModelName": "ActiveListening",
    "Ranges": [
      {
        "LowerBound": null,
        "UpperBound": -0.161,
        "IndexScore": 0,
        "ScoreDescription": "Strongly Negative"
      },

```

```
{
  "LowerBound": -0.161,
  "UpperBound": 4.168,
  "IndexScore": 4,
  "ScoreDescription": "Moderately Negative"
},
{
  "LowerBound": 4.168,
  "UpperBound": 8.332,
  "IndexScore": 5,
  "ScoreDescription": "Neutral"
},
{
  "LowerBound": 8.332,
  "UpperBound": 14.129,
  "IndexScore": 6,
  "ScoreDescription": "Moderately Positive"
},
{
  "LowerBound": 14.129,
  "UpperBound": 44.239,
  "IndexScore": 10,
  "ScoreDescription": "Strongly Positive"
}
]
},
{
  "ModelName": "BeEmpathetic",
  "Ranges": [
    {
      "LowerBound": null,
      "UpperBound": -1.819,
      "IndexScore": 0,
      "ScoreDescription": "Strongly Negative"
    },
    {
      "LowerBound": -1.819,
      "UpperBound": 2.76,
      "IndexScore": 4,
      "ScoreDescription": "Moderately Negative"
    },
    {
      "LowerBound": 2.76,
      "UpperBound": 7.965,
      "IndexScore": 5,
      "ScoreDescription": "Neutral"
    }
  ]
}
```

```
    },
    {
      "LowerBound": 7.965,
      "UpperBound": 15.868,
      "IndexScore": 6,
      "ScoreDescription": "Moderately Positive"
    },
    {
      "LowerBound": 15.868,
      "UpperBound": 92.837,
      "IndexScore": 10,
      "ScoreDescription": "Strongly Positive"
    }
  ]
},
{
  "ModelName": "BuildRapport",
  "Ranges": [
    {
      "LowerBound": null,
      "UpperBound": -3.726,
      "IndexScore": 0,
      "ScoreDescription": "Strongly Negative"
    },
    {
      "LowerBound": -3.726,
      "UpperBound": -1.102,
      "IndexScore": 4,
      "ScoreDescription": "Moderately Negative"
    },
    {
      "LowerBound": -1.102,
      "UpperBound": 1.893,
      "IndexScore": 5,
      "ScoreDescription": "Neutral"
    },
    {
      "LowerBound": 1.893,
      "UpperBound": 8.13,
      "IndexScore": 6,
      "ScoreDescription": "Moderately Positive"
    },
    {
      "LowerBound": 8.13,
      "UpperBound": 76.775,
      "IndexScore": 10,
```



```

        "ScoreDescription": "Strongly Positive"
    }
]
},
{
    "ModelName": "Churn",
    "Ranges": [
        {
            "LowerBound": null,
            "UpperBound": -1.391,
            "IndexScore": 0,
            "ScoreDescription": "Strongly Negative"
        },
        {
            "LowerBound": -1.391,
            "UpperBound": -0.338,
            "IndexScore": 4,
            "ScoreDescription": "Moderately Negative"
        },
        {
            "LowerBound": -0.338,
            "UpperBound": 0.338,
            "IndexScore": 5,
            "ScoreDescription": "Neutral"
        },
        {
            "LowerBound": 0.338,
            "UpperBound": 1.208,
            "IndexScore": 6,
            "ScoreDescription": "Moderately Positive"
        },
        {
            "LowerBound": 1.208,
            "UpperBound": 7.478,
            "IndexScore": 10,
            "ScoreDescription": "Strongly Positive"
        }
    ]
},
{
    "ModelName": "Complaint",
    "Ranges": [
        {
            "LowerBound": null,
            "UpperBound": -6.531,
            "IndexScore": 0,

```

```

        "ScoreDescription": "Strongly Negative"
    },
    {
        "LowerBound": -6.531,
        "UpperBound": -2.926,
        "IndexScore": 4,
        "ScoreDescription": "Moderately Negative"
    },
    {
        "LowerBound": -2.926,
        "UpperBound": -0.29,
        "IndexScore": 5,
        "ScoreDescription": "Neutral"
    },
    {
        "LowerBound": -0.29,
        "UpperBound": 2.879,
        "IndexScore": 6,
        "ScoreDescription": "Moderately Positive"
    },
    {
        "LowerBound": 2.879,
        "UpperBound": 21.566,
        "IndexScore": 10,
        "ScoreDescription": "Strongly Positive"
    }
]
},
{
    "ModelName": "DemonstrateOwnership",
    "Ranges": [
        {
            "LowerBound": null,
            "UpperBound": 0.152,
            "IndexScore": 0,
            "ScoreDescription": "Strongly Negative"
        },
        {
            "LowerBound": 0.152,
            "UpperBound": 3.976,
            "IndexScore": 4,
            "ScoreDescription": "Moderately Negative"
        },
        {
            "LowerBound": 3.976,
            "UpperBound": 7.671,

```

```
    "IndexScore": 5,
    "ScoreDescription": "Neutral"
  },
  {
    "LowerBound": 7.671,
    "UpperBound": 12.841,
    "IndexScore": 6,
    "ScoreDescription": "Moderately Positive"
  },
  {
    "LowerBound": 12.841,
    "UpperBound": 39.623,
    "IndexScore": 10,
    "ScoreDescription": "Strongly Positive"
  }
]
},
{
  "ModelName": "EffectiveQuestioning",
  "Ranges": [
    {
      "LowerBound": null,
      "UpperBound": -0.781,
      "IndexScore": 0,
      "ScoreDescription": "Strongly Negative"
    },
    {
      "LowerBound": -0.781,
      "UpperBound": 3.423,
      "IndexScore": 4,
      "ScoreDescription": "Moderately Negative"
    },
    {
      "LowerBound": 3.423,
      "UpperBound": 7.412,
      "IndexScore": 5,
      "ScoreDescription": "Neutral"
    },
    {
      "LowerBound": 7.412,
      "UpperBound": 13.041,
      "IndexScore": 6,
      "ScoreDescription": "Moderately Positive"
    },
    {
      "LowerBound": 13.041,
```

```

        "UpperBound": 45.808,
        "IndexScore": 10,
        "ScoreDescription": "Strongly Positive"
    }
]
},
{
    "ModelName": "FinancialDistress",
    "Ranges": [
        {
            "LowerBound": null,
            "UpperBound": -2.773,
            "IndexScore": 0,
            "ScoreDescription": "Strongly Negative"
        },
        {
            "LowerBound": -2.773,
            "UpperBound": -1.098,
            "IndexScore": 4,
            "ScoreDescription": "Moderately Negative"
        },
        {
            "LowerBound": -1.098,
            "UpperBound": -0.252,
            "IndexScore": 5,
            "ScoreDescription": "Neutral"
        },
        {
            "LowerBound": -0.252,
            "UpperBound": 0.551,
            "IndexScore": 6,
            "ScoreDescription": "Moderately Positive"
        },
        {
            "LowerBound": 0.551,
            "UpperBound": 6.343,
            "IndexScore": 10,
            "ScoreDescription": "Strongly Positive"
        }
    ]
},
{
    "ModelName": "InappropriateAction",
    "Ranges": [
        {
            "LowerBound": null,

```

```

        "UpperBound": -1.78,
        "IndexScore": 0,
        "ScoreDescription": "Strongly Negative"
    },
    {
        "LowerBound": -1.78,
        "UpperBound": 2.584,
        "IndexScore": 4,
        "ScoreDescription": "Moderately Negative"
    },
    {
        "LowerBound": 2.584,
        "UpperBound": 6.414,
        "IndexScore": 5,
        "ScoreDescription": "Neutral"
    },
    {
        "LowerBound": 6.414,
        "UpperBound": 11.22,
        "IndexScore": 6,
        "ScoreDescription": "Moderately Positive"
    },
    {
        "LowerBound": 11.22,
        "UpperBound": 37.563,
        "IndexScore": 10,
        "ScoreDescription": "Strongly Positive"
    }
]
},
{
    "ModelName": "PromoteSelfService",
    "Ranges": [
        {
            "LowerBound": null,
            "UpperBound": -5.401,
            "IndexScore": 0,
            "ScoreDescription": "Strongly Negative"
        },
        {
            "LowerBound": -5.401,
            "UpperBound": -1.86,
            "IndexScore": 4,
            "ScoreDescription": "Moderately Negative"
        },
    ],
    {

```

```

        "LowerBound": -1.86,
        "UpperBound": 1.757,
        "IndexScore": 5,
        "ScoreDescription": "Neutral"
    },
    {
        "LowerBound": 1.757,
        "UpperBound": 9.594,
        "IndexScore": 6,
        "ScoreDescription": "Moderately Positive"
    },
    {
        "LowerBound": 9.594,
        "UpperBound": 139.409,
        "IndexScore": 10,
        "ScoreDescription": "Strongly Positive"
    }
]
},
{
    "ModelName": "SetExpectations",
    "Ranges": [
        {
            "LowerBound": null,
            "UpperBound": -3.394,
            "IndexScore": 0,
            "ScoreDescription": "Strongly Negative"
        },
        {
            "LowerBound": -3.394,
            "UpperBound": 0.137,
            "IndexScore": 4,
            "ScoreDescription": "Moderately Negative"
        },
        {
            "LowerBound": 0.137,
            "UpperBound": 3.358,
            "IndexScore": 5,
            "ScoreDescription": "Neutral"
        },
        {
            "LowerBound": 3.358,
            "UpperBound": 7.618,
            "IndexScore": 6,
            "ScoreDescription": "Moderately Positive"
        }
    ],

```

```

    {
      "LowerBound": 7.618,
      "UpperBound": 24.802,
      "IndexScore": 10,
      "ScoreDescription": "Strongly Positive"
    }
  ]
},
{
  "ModelName": "PromoteSelfService",
  "Ranges": [
    {
      "LowerBound": null,
      "UpperBound": -5.401,
      "IndexScore": 0,
      "ScoreDescription": "Strongly Negative"
    },
    {
      "LowerBound": -5.401,
      "UpperBound": -1.86,
      "IndexScore": 4,
      "ScoreDescription": "Moderately Negative"
    },
    {
      "LowerBound": -1.86,
      "UpperBound": 1.757,
      "IndexScore": 5,
      "ScoreDescription": "Neutral"
    },
    {
      "LowerBound": 1.757,
      "UpperBound": 9.594,
      "IndexScore": 6,
      "ScoreDescription": "Moderately Positive"
    },
    {
      "LowerBound": 9.594,
      "UpperBound": 139.409,
      "IndexScore": 10,
      "ScoreDescription": "Strongly Positive"
    }
  ]
},
{
  "ModelName": "Updated",
  "Ranges": [

```

```
{
  "LowerBound": null,
  "UpperBound": -0.31,
  "IndexScore": 0,
  "ScoreDescription": "Strongly Negative"
},
{
  "LowerBound": -0.31,
  "UpperBound": 2.489,
  "IndexScore": 4,
  "ScoreDescription": "Moderately Negative"
},
{
  "LowerBound": 2.489,
  "UpperBound": 5.538,
  "IndexScore": 5,
  "ScoreDescription": "Neutral"
},
{
  "LowerBound": 5.538,
  "UpperBound": 8.988,
  "IndexScore": 6,
  "ScoreDescription": "Moderately Positive"
},
{
  "LowerBound": 8.988,
  "UpperBound": 25.962,
  "IndexScore": 10,
```

Sales Effectiveness Model Scores

 [SalesEffectivenessModelScores_v2.json](#)




```
[
  {
    "ModelName": "Escalation",
    "Ranges": [
      {
        "LowerBound": null,
        "UpperBound": -0.034,
        "IndexScore": 0,
        "ScoreDescription": "Strongly Negative"
      },
      {
        "LowerBound": -0.034,
        "UpperBound": 0.009,
        "IndexScore": 4,
        "ScoreDescription": "Moderately Negative"
      },
      {
        "LowerBound": 0.009,
        "UpperBound": 0.148,
        "IndexScore": 5,
        "ScoreDescription": "Neutral"
      },
      {
        "LowerBound": 0.148,
        "UpperBound": 0.511,
        "IndexScore": 6,
        "ScoreDescription": "Moderately Positive"
      },
      {
        "LowerBound": 0.511,
        "UpperBound": 31.828,
        "IndexScore": 10,
        "ScoreDescription": "Strongly Positive"
      }
    ]
  },
  {
    "ModelName": "Resolution",
    "Ranges": [
      {
        "LowerBound": null,
        "UpperBound": -1.612,
        "IndexScore": 0,
        "ScoreDescription": "Strongly Negative"
      },

```

```
{
  "LowerBound": -1.612,
  "UpperBound": -0.08,
  "IndexScore": 4,
  "ScoreDescription": "Moderately Negative"
},
{
  "LowerBound": -0.08,
  "UpperBound": 1.454,
  "IndexScore": 5,
  "ScoreDescription": "Neutral"
},
{
  "LowerBound": 1.454,
  "UpperBound": 6.443,
  "IndexScore": 6,
  "ScoreDescription": "Moderately Positive"
},
{
  "LowerBound": 6.443,
  "UpperBound": 25.094,
  "IndexScore": 10,
  "ScoreDescription": "Strongly Positive"
}
]
},
{
  "ModelName": "Sentiment",
  "Ranges": [
    {
      "LowerBound": null,
      "UpperBound": -0.191,
      "IndexScore": 0,
      "ScoreDescription": "Strongly Negative"
    },
    {
      "LowerBound": -0.191,
      "UpperBound": -0.14,
      "IndexScore": 4,
      "ScoreDescription": "Moderately Negative"
    },
    {
      "LowerBound": -0.14,
      "UpperBound": 2.284,
      "IndexScore": 5,
      "ScoreDescription": "Neutral"
    }
  ]
}
```

```
    },
    {
      "LowerBound": 2.284,
      "UpperBound": 5.471,
      "IndexScore": 6,
      "ScoreDescription": "Moderately Positive"
    },
    {
      "LowerBound": 5.471,
      "UpperBound": 14.715,
      "IndexScore": 10,
      "ScoreDescription": "Strongly Positive"
    }
  ]
},
{
  "ModelName": "AcknowledgeRequest",
  "Ranges": [
    {
      "LowerBound": null,
      "UpperBound": -20.964,
      "IndexScore": 0,
      "ScoreDescription": "Strongly Negative"
    },
    {
      "LowerBound": -20.964,
      "UpperBound": -9.675,
      "IndexScore": 4,
      "ScoreDescription": "Moderately Negative"
    },
    {
      "LowerBound": -9.675,
      "UpperBound": -0.95,
      "IndexScore": 5,
      "ScoreDescription": "Neutral"
    },
    {
      "LowerBound": -0.95,
      "UpperBound": 0.218,
      "IndexScore": 6,
      "ScoreDescription": "Moderately Positive"
    },
    {
      "LowerBound": 0.218,
      "UpperBound": 13.948,
      "IndexScore": 10,
```

```

        "ScoreDescription": "Strongly Positive"
    }
]
},
{
    "ModelName": "AskForTheSale",
    "Ranges": [
        {
            "LowerBound": null,
            "UpperBound": -1.509,
            "IndexScore": 0,
            "ScoreDescription": "Strongly Negative"
        },
        {
            "LowerBound": -1.509,
            "UpperBound": 0,
            "IndexScore": 4,
            "ScoreDescription": "Moderately Negative"
        },
        {
            "LowerBound": 0,
            "UpperBound": 0.968,
            "IndexScore": 5,
            "ScoreDescription": "Neutral"
        },
        {
            "LowerBound": 0.968,
            "UpperBound": 6.204,
            "IndexScore": 6,
            "ScoreDescription": "Moderately Positive"
        },
        {
            "LowerBound": 6.204,
            "UpperBound": 38.935,
            "IndexScore": 10,
            "ScoreDescription": "Strongly Positive"
        }
    ]
},
{
    "ModelName": "ConfirmedSale",
    "Ranges": [
        {
            "LowerBound": null,
            "UpperBound": -6.792,
            "IndexScore": 0,

```

```
    "ScoreDescription": "Strongly Negative"
  },
  {
    "LowerBound": -6.792,
    "UpperBound": -1.514,
    "IndexScore": 4,
    "ScoreDescription": "Moderately Negative"
  },
  {
    "LowerBound": -1.514,
    "UpperBound": 0.071,
    "IndexScore": 5,
    "ScoreDescription": "Neutral"
  },
  {
    "LowerBound": 0.071,
    "UpperBound": 7.945,
    "IndexScore": 6,
    "ScoreDescription": "Moderately Positive"
  },
  {
    "LowerBound": 7.945,
    "UpperBound": 50.349,
    "IndexScore": 10,
    "ScoreDescription": "Strongly Positive"
  }
]
},
{
  "ModelName": "DemonstrateEmpathy",
  "Ranges": [
    {
      "LowerBound": null,
      "UpperBound": -0.164,
      "IndexScore": 0,
      "ScoreDescription": "Strongly Negative"
    },
    {
      "LowerBound": -0.164,
      "UpperBound": 0.744,
      "IndexScore": 4,
      "ScoreDescription": "Moderately Negative"
    },
    {
      "LowerBound": 0.744,
      "UpperBound": 4.476,
```

```
    "IndexScore": 5,
    "ScoreDescription": "Neutral"
  },
  {
    "LowerBound": 4.476,
    "UpperBound": 11.334,
    "IndexScore": 6,
    "ScoreDescription": "Moderately Positive"
  },
  {
    "LowerBound": 11.334,
    "UpperBound": 41.605,
    "IndexScore": 10,
    "ScoreDescription": "Strongly Positive"
  }
]
},
{
  "ModelName": "DemonstrateOwnership",
  "Ranges": [
    {
      "LowerBound": null,
      "UpperBound": -0.15,
      "IndexScore": 0,
      "ScoreDescription": "Strongly Negative"
    },
    {
      "LowerBound": -0.15,
      "UpperBound": 0.783,
      "IndexScore": 4,
      "ScoreDescription": "Moderately Negative"
    },
    {
      "LowerBound": 0.783,
      "UpperBound": 4.305,
      "IndexScore": 5,
      "ScoreDescription": "Neutral"
    },
    {
      "LowerBound": 4.305,
      "UpperBound": 11.842,
      "IndexScore": 6,
      "ScoreDescription": "Moderately Positive"
    },
    {
      "LowerBound": 11.842,
```



```

        "UpperBound": 39.685,
        "IndexScore": 10,
        "ScoreDescription": "Strongly Positive"
    }
]
},
{
    "ModelName": "LikelihoodToBuy",
    "Ranges": [
        {
            "LowerBound": null,
            "UpperBound": -2.495,
            "IndexScore": 0,
            "ScoreDescription": "Strongly Negative"
        },
        {
            "LowerBound": -2.495,
            "UpperBound": -0.09,
            "IndexScore": 4,
            "ScoreDescription": "Moderately Negative"
        },
        {
            "LowerBound": -0.09,
            "UpperBound": 1.083,
            "IndexScore": 5,
            "ScoreDescription": "Neutral"
        },
        {
            "LowerBound": 1.083,
            "UpperBound": 5.105,
            "IndexScore": 6,
            "ScoreDescription": "Moderately Positive"
        },
        {
            "LowerBound": 5.105,
            "UpperBound": 30.162,
            "IndexScore": 10,
            "ScoreDescription": "Strongly Positive"
        }
    ]
},
{
    "ModelName": "MakeAConnection",
    "Ranges": [
        {
            "LowerBound": null,

```

```

        "UpperBound": -2.094,
        "IndexScore": 0,
        "ScoreDescription": "Strongly Negative"
    },
    {
        "LowerBound": -2.094,
        "UpperBound": -0.032,
        "IndexScore": 4,
        "ScoreDescription": "Moderately Negative"
    },
    {
        "LowerBound": -0.032,
        "UpperBound": 3.395,
        "IndexScore": 5,
        "ScoreDescription": "Neutral"
    },
    {
        "LowerBound": 3.395,
        "UpperBound": 13.156,
        "IndexScore": 6,
        "ScoreDescription": "Moderately Positive"
    },
    {
        "LowerBound": 13.156,
        "UpperBound": 67.959,
        "IndexScore": 10,
        "ScoreDescription": "Strongly Positive"
    }
]
},
{
    "ModelName": "OvercomeObjections",
    "Ranges": [
        {
            "LowerBound": null,
            "UpperBound": -0.26,
            "IndexScore": 0,
            "ScoreDescription": "Strongly Negative"
        },
        {
            "LowerBound": -0.26,
            "UpperBound": -0.044,
            "IndexScore": 4,
            "ScoreDescription": "Moderately Negative"
        },
    ],
    {

```

```
    "LowerBound": -0.044,  
    "UpperBound": 0.076,  
    "IndexScore": 5,  
    "ScoreDescription": "Neutral"  
  },  
  {  
    "LowerBound": 0.076,  
    "UpperBound": 0.585,  
    "IndexScore": 6,  
    "ScoreDescription": "Moderately Positive"  
  },  
  {  
    "LowerBound": 0.585,  
    "UpperBound": 8.479,  
    "IndexScore": 10,  
    "ScoreDescription": "Strongly Positive"  
  }  
]  
},  
{  
  "ModelName": "PresumptiveSelling",  
  "Ranges": [  
    {  
      "LowerBound": null,  
      "UpperBound": -0.071,  
      "IndexScore": 0,  
      "ScoreDescription": "Strongly Negative"  
    },  
    {  
      "LowerBound": -0.071,  
      "UpperBound": -0.006,  
      "IndexScore": 4,  
      "ScoreDescription": "Moderately Negative"  
    },  
    {  
      "LowerBound": -0.006,  
      "UpperBound": 0.047,  
      "IndexScore": 5,  
      "ScoreDescription": "Neutral"  
    },  
    {  
      "LowerBound": 0.047,  
      "UpperBound": 0.41,  
      "IndexScore": 6,  
      "ScoreDescription": "Moderately Positive"  
    }  
  ],  
}
```


7. Generative AI

7.1. ElevateAI Generative AI API calls

// Overview

Welcome to the ElevateAI Generative AI API documentation. This API provides various endpoints to access Generative AI features for analyzing interactions, generating extensive summaries, extracting insights, and facilitating agent coaching.

Each of these endpoints utilize the unique `interactionIdentifier` returned upon successful declaration of the `audio`, `transcript`, or `chat` interaction.

There are two categories of Generative AI endpoints available for interactions. They are Contact Center – or Customer Experience (CX) – specific and Generic. **CX**-specific endpoints expect the interaction to be of a format typical of a contact center, e.g. agent and customer. **Generic** endpoints anticipate a more generalized interaction structure between the parties.

There is a limit of 10 concurrent Generative AI requests at a time, per account. Requests beyond 10 will return the response documented [here](#).

// Prerequisites

Audio interactions **must have completed processing** (see [Check the Processing Status](#)) before the Generative AI Endpoints can provide successful responses. Attempting to make a request prior to the completed processing of an audio interaction will result in an error response.

The Generative AI requests are available immediately upon declaration of transcript and chat interactions.

7.2. Making API Requests for Generative AI Results

All of the ElevateAI Generative AI API Endpoints use a similar format, leveraging the unique `interactionIdentifier` returned upon successfully declaring the [audio](#), [transcript](#), or [chat](#) interaction.

When the desired `GET` request* successfully executes, an `HTTP` status is returned to indicate the request was successful, along with a `JSON` response providing details specific to that endpoint. Details specific to an individual endpoint are covered on the [Generative AI API Endpoint Details](#) page.

* All endpoints support a GET method. The *Ask ElevateAI* endpoint supports both POST and GET methods. See the documentation for that endpoint for details.

The general request format to be used for all of the Generative AI API requests is as follows:

General Request Format

Retrieve Generative AI Response

GET

https://api.elevateai.com/v1/interactions/{interactionIdentifier} 

Request

PATH PARAMS

interactionIdentifier String **required**

interactionIdentifier returned upon successfully declaring the interaction

endpoint String **required**

The specific API endpoint for the desired Generative AI response. Endpoints are covered on the Generative AI API Endpoint Details page.

HEADER PARAMETERS

X-API-Token String **required**

Valid API key associated with your account for authentication and usage tracking

Accept-Encoding String **optional**

Recommended for bandwidth optimization, will be 'gzip, deflate, br'

Accept-Language String **optional**

To be used for the Summary endpoint only (ignored for other endpoints). Specify the language code for the desired language of the response (e.g. pt-BR). Default is English.



Curl



Node.js



JS



Python



Ruby



```
var request = require('request');
var options = {
  'method': 'GET',
  'url': 'https://api.elevateai.com/v1/interactions/{interactionIdentifie
  'headers': {
    'Accept': 'application/json',
    'Content-Type': 'application/json'
  }
};
request(options, function (error, response) {
  if (error) throw new Error(error);
  console.log(response.body);
});
```

Responses

● 200 ● 404 ● 408 ● 409 ● 422 ● 429 ● 500 ● 504



```
{
  "errorMessage (string)"
}
```

Retryable Responses

The following response codes leave the interaction endpoint in a retryable state.

Calling code should handle these responses gracefully and retry appropriately:

Response	Message	Meaning
408	RequestTimeout	The Generative AI model was not able to respond in an acceptable amount of time.
429	TooManyRequests	By default, each account is allowed up to 10 simultaneous Generative AI requests at one time. This response indicates that more than 10 are being attempted.

7.3. Generative AI API Endpoint Details

Each of the Generative AI endpoints utilize the unique `interactionIdentifier` returned upon successful declaration of the `audio`, `transcript`, or `chat` interaction.

NOTE: `Audio` interactions must have completed processing before the Generative AI Endpoints can provide success responses. Attempting to make a request prior to the completed processing of an `audio` interaction will result in an error response.

Chats and Transcripts allow for immediate access to Generative AI endpoints after declaration.

The **Generative AI REST API** calls are all similar in format. For details on that format, please see the [Making API Requests for Generative AI Results](#) page. Unsuccessful response codes are documented on that page, as well.

Access a guide to using these features below.

// CX Generative AI API Features

AutoSummary

- **Description:** Retrieves a summary of the interaction between the Agent and Customer
- **Endpoint:** `/v1/interactions/{interactionIdentifier:guid}/gen-ai/cx/summary`

Successful Response



```
{
  "summary": string
}
```

Summary Details

- **Description:** Summarizes additional details about the interaction, including the reason(s) for the interaction, actions taken, entities mentioned, and whether or not

a resolution was reached

- **Endpoint:** /v1/interactions/{interactionIdentifier:guid}/[gen-ai/cx/summary/details](#)

Successful Response



```
{
  "reasons": string[],
  "actionsTakenByAgent": string[],
  "entities": [{
    "type": string,
    "value": string
  }],
  "isResolved": boolean
}
```

Agent Action Items

- **Description:** Provides insight into the specific actions taken by an agent over the course of a customer interaction
- **Endpoint:** /v1/interactions/{interactionIdentifier:guid}/[gen-ai/cx/agent/action-items](#)

Successful Response



```
[{
  "action": string,
  "details": string
}]
```

Agent Coaching Assistant

- **Description:** Provides insight into agent behaviors throughout an interaction, allowing users to identify opportunities for agent coaching and/or agent improvement after a customer interaction
- **Endpoint:** /v1/interactions/{interactionIdentifier:guid}/[gen-ai/cx/agent/coaching-assistant](#)

Successful Response



```
{
  "coaching": string
}
```

Interaction Topics

- **Description:** Provides a list of the topics that were discussed during the interaction
- **Endpoint:** /v1/interactions/{interactionIdentifier:guid}/[gen-ai/cx/topics](#)

Successful Response



```
[{
  "topic": string,
  "description": string
}]
```

Conversation Composition

- **Description:** Provides a list of the questions and/or concerns that were raised by the customer during an interaction, the steps used to address and/or resolve each issue, and the status of each issue, post-interaction
- **Endpoint:** /v1/interactions/{interactionIdentifier:guid}/[gen-ai/cx/conversation-composition](#)

Successful Response



```
[{
  "concernSummary": string,
  "concern": string,
  "isResolved": boolean,
  "actionsTakenByAgent": string[]
}]
```

Ask ElevateAI Q&A

- **Description:** Allows the user to ask questions around a specific interaction – up to 5 questions per interaction – and receive specific answers.
 - The data being queried is your data!
 - A single question – up to 2000 characters – is asked in the request body of the **POST** method. The answer to the question is returned in the response.
 - Up to a maximum of 5 successful questions or 5 error responses are allowed per interaction.
 - This endpoint also supports a **GET** method which returns all of the prior question & answer pairs for this interaction (including failed questions and responses)
- **Endpoint:** /v1/interactions/{interactionIdentifier:guid}/gen-ai/cx/ask

POST Request Body

POST Successful Response

GET Successful Response



```
{
  "question": "<enter your question here>"
}
```

// Generic (non-CX) Generative AI API Features

Summary

- **Description:** Retrieves a summary of the interaction
- **Endpoint:** /v1/interactions/{interactionIdentifier:guid}/gen-ai/summary

Successful Response



```
{
  "summary": string
}
```

Summary Details

- **Description:** Summarizes the entities mentioned within the interaction

- **Endpoint:** /v1/interactions/{interactionIdentifier:guid}/[gen-ai/summary/details](#)

Successful Response



```
{
  "entities": [{
    "type": string,
    "value": string
  }]
}
```

Action Items

- **Description:** Describes the action items which were raised during the interaction
- **Endpoint:** /v1/interactions/{interactionIdentifier:guid}/[gen-ai/action-items](#)

Successful Response



```
[{
  "action": string,
  "details": string
}]
```

Topics

- **Description:** Returns a list of topics that were discussed during the interaction
- **Endpoint:** /v1/interactions/{interactionIdentifier:guid}/[gen-ai/topics](#)

Successful Response



```
[{
  "topic": string,
  "description": string
}]
```

Ask ElevateAI

- **Description:** Allows the user to ask questions around a specific interaction – up to 5 questions per interaction – and receive specific answers.
 - The data being queried is your data!
 - A single question – up to 2000 characters – is asked in the request body of the **POST** method. The answer to the question is returned in the response.
 - Up to a maximum of 5 successful questions or 5 error responses are allowed per interaction.
 - This endpoint also supports a **GET** method which returns all of the prior question & answer pairs for this interaction (including failed questions and responses).
- **Endpoint:** /v1/interactions/{interactionIdentifier:guid}/[gen-ai/ask](#)

POST Request Body

POST Successful Response

GET Successful Response



```
{  
  "question": "<enter your question here>"  
}
```

For any further assistance or inquiries, please contact our support team [here](#).

8. Delete Data

// Overview

Leveraging the unique `interactionIdentifier` provided upon declaring an `audio`, `chat`, or `transcript` interaction, this endpoint enables you to **instantly delete all results** associated with the interaction from our database.

Note: All source data is deleted immediately and promptly upon processing. See [Data Retention & Security](#) for additional details.

When the `DELETE` request **successfully executes**, an HTTP status is returned to indicate the request was successful and all data associated with the `interactionIdentifier` has been removed from our database.

If an error occurs when requesting to declare the interaction, a standard HTTP response code is returned to indicate the request was unsuccessful, along with a JSON response containing additional details to assist in [troubleshooting](#).

// Request Parameters & Code Example

Delete All Data