

Starting April 29, 2025, Gemini 1.5 Pro and Gemini 1.5 Flash models are not available in projects that have no prior usage of these models, including new projects. For details, see [Model versions and lifecycle](#) (/vertex-ai/generative-ai/docs/learn/model-versions#legacy-stable).

# Live API reference

## Preview

This feature is subject to the "Pre-GA Offerings Terms" in the General Service Terms section of the [Service Specific Terms](#) (/terms/service-terms#1). Pre-GA features are available "as is" and might have limited support. For more information, see the [launch stage descriptions](#) (/products#product-launch-stages).

To try a tutorial that lets you use your voice and camera to talk to Gemini through the Live API, see the [websocket-demo-app tutorial](#) (<https://github.com/GoogleCloudPlatform/generative-ai/tree/main/gemini/multimodal-live-api/websocket-demo-app>).

The Live API enables low-latency bidirectional voice and video interactions with Gemini. Using the Live API, you can provide end users with the experience of natural, human-like voice conversations, and with the ability to interrupt the model's responses using voice commands. The Live API can process text, audio, and video input, and it can provide text and audio output.

---

## Multimodal live API demo



For more information about the Live API, see [Live API](#) (/vertex-ai/generative-ai/docs/live-api).

## Capabilities

Live API includes the following key capabilities:



- **Multimodality:** The model can see, hear, and speak.

- **Low-latency realtime interaction:** The model can provide fast responses.
- **Session memory:** The model retains memory of all interactions within a single session, recalling previously heard or seen information.
- **Support for function calling, code execution, and Search as a Tool:** You can integrate the model with external services and data sources.

Live API is designed for server-to-server communication.

For web and mobile apps, we recommend using the integration from our partners at [Daily](https://www.daily.co/products/gemini/multimodal-live-api/) (<https://www.daily.co/products/gemini/multimodal-live-api/>).

## Supported models

- [Gemini 2.5 Flash with Live API native Audio](#)  
(</vertex-ai/generative-ai/docs/models/gemini/2-5-flash#live-api-native-audio>) 
- [Gemini 2.0 Flash with Live API](#) (</vertex-ai/generative-ai/docs/models/gemini/2-0-flash>) 

## Get started

To try the Live API, go to the [Vertex AI Studio](#) (<https://console.cloud.google.com/vertex-ai/studio/multimodal-live>), and then click **Start Session**.

Live API is a stateful API that uses [WebSockets](#) (<https://en.wikipedia.org/wiki/WebSocket>).

This section shows an example of how to use Live API for text-to-text generation, using Python 3.9+.

[Gen AI SDK for Python](#)  
([#gen-ai-sdk-for-python](#))

```
# Replace the `GOOGLE_CLOUD_PROJECT` and `GOOGLE_CLOUD_LOCATION` values
# with appropriate values for your project.

from google import genai
from google.genai.types import (
    Content,
    LiveConnectConfig,
    Modality,
    Part,
```

```

)

client = genai.Client(
    vertexai=True,
    project=GOOGLE_CLOUD_PROJECT,
    location=GOOGLE_CLOUD_LOCATION
)
MODEL_ID = "gemini-2.0-flash-live-preview-04-09"

async with client.aio.live.connect(
    model=MODEL_ID,
    config=LiveConnectConfig(response_modalities=[Modality.TEXT]),
) as session:
    text_input = "Hello? Gemini, are you there?"
    print("> ", text_input, "\n")
    await session.send_client_content(
        turns=Content(role="user", parts=[Part(text=text_input)])
    )

    response = []

    async for message in session.receive():
        if message.text:
            response.append(message.text)

    print("".join(response))
# Example output:
# > Hello? Gemini, are you there?
# Yes, I'm here. What would you like to talk about?

```

## Integration guide

This section describes how integration works with Live API.

### Sessions

A WebSocket connection establishes a session between the client and the Gemini server.

After a client initiates a new connection the session can exchange messages with the server to:

- Send text, audio, or video to the Gemini server.

- Receive audio, text, or function call requests from the Gemini server.

The session configuration is sent in the first message after connection. A session configuration includes the model, generation parameters, system instructions, and tools.

See the following example configuration:

```
{
  "model": string,
  "generationConfig": {
    "candidateCount": integer,
    "maxOutputTokens": integer,
    "temperature": number,
    "topP": number,
    "topK": integer,
    "presencePenalty": number,
    "frequencyPenalty": number,
    "responseModalities": [string],
    "speechConfig": object
  },
  "systemInstruction": string,
  "tools": [object]
}
```

For more information, see [BidiGenerateContentSetup](#) (#bidigeneratecontentsetup).

## Send messages

Messages are JSON-formatted objects exchanged over the WebSocket connection.

To send a message the client must send a JSON object over an open WebSocket connection. The JSON object must have *exactly one* of the fields from the following object set:

```
{
  "setup": BidiGenerateContentSetup,
  "clientContent": BidiGenerateContentClientContent,
  "realtimeInput": BidiGenerateContentRealtimeInput,
  "toolResponse": BidiGenerateContentToolResponse
}
```

```
}
```

Supported client messages

See the supported client messages in the following table:

Message	Description
<b>BidiGenerateContentSetup</b>	Session configuration to be sent in the first message
<b>BidiGenerateContentClientContent</b>	Incremental content update of the current conversation delivered from the client
<b>BidiGenerateContentRealtimeInput</b>	Real time audio or video input
<b>BidiGenerateContentToolResponse</b>	Response to a <b>ToolCallMessage</b> received from the server

Receive messages

To receive messages from Gemini, listen for the WebSocket 'message' event, and then parse the result according to the definition of the supported server messages.

See the following:

```
ws.addEventListener("message", async (evt) => {
  if (evt.data instanceof Blob) {
    // Process the received data (audio, video, etc.)
  } else {
    // Process JSON response
  }
});
```

Server messages will have *exactly one* of the fields from the following object set:

```
{
  "setupComplete": BidiGenerateContentSetupComplete,
  "serverContent": BidiGenerateContentServerContent,
  "toolCall": BidiGenerateContentToolCall,
```

```
"toolCallCancellation": BidiGenerateContentToolCallCancellation
"usageMetadata": UsageMetadata
"goAway": GoAway
"sessionResumptionUpdate": SessionResumptionUpdate
"inputTranscription": BidiGenerateContentTranscription
"outputTranscription": BidiGenerateContentTranscription
}
```

Supported server messages

See the supported server messages in the following table:

Message	Description
BidiGenerateContentSetupComplete	A <b>BidiGenerateContentSetup</b> message from the client, sent when setup is complete
BidiGenerateContentServerContent	Content generated by the model in response to a client message
BidiGenerateContentToolCall	Request for the client to run the function calls and return the responses with the matching IDs
BidiGenerateContentToolCallCancellation	Sent when a function call is canceled due to the user interrupting model output
UsageMetadata	A report of the number of tokens used by the session so far
GoAway	A signal that the current connection will soon be terminated
SessionResumptionUpdate	A session checkpoint, which can be resumed
BidiGenerateContentTranscription	A transcription of either the user's or model's speech

Incremental content updates

Use incremental updates to send text input, establish session context, or restore session context. For short contexts you can send turn-by-turn interactions to represent the exact sequence of events. For longer contexts it's recommended to provide a single message summary to free up the context window for the follow up interactions.

See the following example context message:

```
{
  "clientContent": {
    "turns": [
      {
        "parts": [
          {
            "text": ""
          }
        ],
        "role": "user"
      },
      {
        "parts": [
          {
            "text": ""
          }
        ],
        "role": "model"
      }
    ],
    "turnComplete": true
  }
}
```

Note that while content parts can be of a `functionResponse` type, `BidiGenerateContentClientContent` shouldn't be used to provide a response to the function calls issued by the model. `BidiGenerateContentToolResponse` should be used instead. `BidiGenerateContentClientContent` should only be used to establish previous context or provide text input to the conversation.

## Streaming audio and video

To see an example of how to use the Live API in a streaming audio and video format, run the "Getting started with the Multimodal Live API" Jupyter notebook in one of the following environments:

[Open in Colab](#)

([https://colab.research.google.com/github/GoogleCloudPlatform/generative-ai/blob/main/gemini/multimodal-live-api/intro\\_multimodal\\_live\\_api\\_genai\\_sdk.ipynb](https://colab.research.google.com/github/GoogleCloudPlatform/generative-ai/blob/main/gemini/multimodal-live-api/intro_multimodal_live_api_genai_sdk.ipynb))

| [Open in Colab Enterprise](#)

([https://console.cloud.google.com/vertex-ai/colab/import/https%3A%2F%2Fraw.githubusercontent.com%2FGoogleCloudPlatform%2Fgenerative-ai%2Fmain%2Fgemini%2Fmultimodal-live-api%2Fintro\\_multimodal\\_live\\_api\\_genai\\_sdk.ipynb](https://console.cloud.google.com/vertex-ai/colab/import/https%3A%2F%2Fraw.githubusercontent.com%2FGoogleCloudPlatform%2Fgenerative-ai%2Fmain%2Fgemini%2Fmultimodal-live-api%2Fintro_multimodal_live_api_genai_sdk.ipynb))

| [Open in Vertex AI Workbench user-managed notebooks](#)

([https://console.cloud.google.com/vertex-ai/workbench/deploy-notebook?download\\_url=https%3A%2F%2Fraw.githubusercontent.com%2FGoogleCloudPlatform%2Fgenerative-ai%2Fmain%2Fgemini%2Fmultimodal-live-api%2Fintro\\_multimodal\\_live\\_api\\_genai\\_sdk.ipynb](https://console.cloud.google.com/vertex-ai/workbench/deploy-notebook?download_url=https%3A%2F%2Fraw.githubusercontent.com%2FGoogleCloudPlatform%2Fgenerative-ai%2Fmain%2Fgemini%2Fmultimodal-live-api%2Fintro_multimodal_live_api_genai_sdk.ipynb))

| [View on GitHub](#)

([https://github.com/GoogleCloudPlatform/generative-ai/blob/main/gemini/multimodal-live-api/intro\\_multimodal\\_live\\_api\\_genai\\_sdk.ipynb](https://github.com/GoogleCloudPlatform/generative-ai/blob/main/gemini/multimodal-live-api/intro_multimodal_live_api_genai_sdk.ipynb))

## Code execution

To see an example of code execution, run the "Intro to Generating and Executing Python Code with Gemini 2.0" Jupyter notebook in one of the following environments:

[Open in Colab](#)

([https://colab.research.google.com/github/GoogleCloudPlatform/generative-ai/blob/main/gemini/code-execution/intro\\_code\\_execution.ipynb](https://colab.research.google.com/github/GoogleCloudPlatform/generative-ai/blob/main/gemini/code-execution/intro_code_execution.ipynb))

| [Open in Colab Enterprise](#)

([https://console.cloud.google.com/vertex-ai/colab/import/https%3A%2F%2Fraw.githubusercontent.com%2FGoogleCloudPlatform%2Fgenerative-ai%2Fmain%2Fgemini%2Fcode-execution%2Fintro\\_code\\_execution.ipynb](https://console.cloud.google.com/vertex-ai/colab/import/https%3A%2F%2Fraw.githubusercontent.com%2FGoogleCloudPlatform%2Fgenerative-ai%2Fmain%2Fgemini%2Fcode-execution%2Fintro_code_execution.ipynb))

| [Open in Vertex AI Workbench user-managed notebooks](#)

([https://console.cloud.google.com/vertex-ai/workbench/deploy-notebook?download\\_url=https%3A%2F%2Fraw.githubusercontent.com%2FGoogleCloudPlatform%2Fgenerative-ai%2Fmain%2Fgemini%2Fcode-execution%2Fintro\\_code\\_execution.ipynb](https://console.cloud.google.com/vertex-ai/workbench/deploy-notebook?download_url=https%3A%2F%2Fraw.githubusercontent.com%2FGoogleCloudPlatform%2Fgenerative-ai%2Fmain%2Fgemini%2Fcode-execution%2Fintro_code_execution.ipynb))

| [View on GitHub](#)

([https://github.com/GoogleCloudPlatform/generative-ai/blob/main/gemini/code-execution/intro\\_code\\_execution.ipynb](https://github.com/GoogleCloudPlatform/generative-ai/blob/main/gemini/code-execution/intro_code_execution.ipynb))

To learn more about code execution, see [Code execution](#)

(</vertex-ai/generative-ai/docs/multimodal/code-execution>).

## Function calling

To see an example of function calling, run the "Intro to Function Calling with the Gemini API" Jupyter notebook in one of the following environments:



### Open in Colab

([https://colab.research.google.com/github/GoogleCloudPlatform/generative-ai/blob/main/gemini/function-calling/intro\\_function\\_calling.ipynb](https://colab.research.google.com/github/GoogleCloudPlatform/generative-ai/blob/main/gemini/function-calling/intro_function_calling.ipynb))

### | Open in Colab Enterprise

([https://console.cloud.google.com/vertex-ai/colab/import/https%3A%2F%2Fraw.githubusercontent.com%2FGoogleCloudPlatform%2Fgenerative-ai%2Fmain%2Fgemini%2Ffunction-calling%2Fintro\\_function\\_calling.ipynb](https://console.cloud.google.com/vertex-ai/colab/import/https%3A%2F%2Fraw.githubusercontent.com%2FGoogleCloudPlatform%2Fgenerative-ai%2Fmain%2Fgemini%2Ffunction-calling%2Fintro_function_calling.ipynb))

### | Open in Vertex AI Workbench user-managed notebooks

([https://console.cloud.google.com/vertex-ai/workbench/deploy-notebook?download\\_url=https%3A%2F%2Fraw.githubusercontent.com%2FGoogleCloudPlatform%2Fgenerative-ai%2Fmain%2Fgemini%2Ffunction-calling%2Fintro\\_function\\_calling.ipynb](https://console.cloud.google.com/vertex-ai/workbench/deploy-notebook?download_url=https%3A%2F%2Fraw.githubusercontent.com%2FGoogleCloudPlatform%2Fgenerative-ai%2Fmain%2Fgemini%2Ffunction-calling%2Fintro_function_calling.ipynb))

### | View on GitHub

([https://github.com/GoogleCloudPlatform/generative-ai/blob/main/gemini/function-calling/intro\\_function\\_calling.ipynb](https://github.com/GoogleCloudPlatform/generative-ai/blob/main/gemini/function-calling/intro_function_calling.ipynb))

All functions must be declared at the start of the session by sending tool definitions as part of the `BidiGenerateContentSetup` message.

You define functions by using JSON, specifically with a [select subset](#)

(<https://ai.google.dev/api/caching#schema>) of the [OpenAPI schema format](#)

(<https://spec.openapis.org/oas/v3.0.3#schema>). A single function declaration can include the following parameters:

- **name** (string): The unique identifier for the function within the API call.
- **description** (string): A comprehensive explanation of the function's purpose and capabilities.
- **parameters** (object): Defines the input data required by the function.
  - **type** (string): Specifies the overall data type, such as object.
  - **properties** (object): Lists individual parameters, each with:
    - **type** (string): The data type of the parameter, such as string, integer, boolean.
    - **description** (string): A clear explanation of the parameter's purpose and expected format.
  - **required** (array): An array of strings listing the parameter names that are mandatory for the function to operate.

For code examples of a function declaration using curl commands, see [Function calling with the Gemini API](#) (<https://ai.google.dev/gemini-api/docs/function-calling#function-calling-curl-samples>). For examples of how to create function declarations using the Gemini API SDKs, see the [Function calling tutorial](#) (<https://ai.google.dev/gemini-api/docs/function-calling/tutorial>).

From a single prompt, the model can generate multiple function calls and the code necessary to chain their outputs. This code executes in a sandbox environment, generating subsequent `BidiGenerateContentToolCall` messages. The execution pauses until the results of each function call are available, which ensures sequential processing.

The client should respond with `BidiGenerateContentToolResponse`.

To learn more, see [Introduction to function calling](/vertex-ai/generative-ai/docs/multimodal/function-calling) (</vertex-ai/generative-ai/docs/multimodal/function-calling>).

## Audio formats

See the list of [supported audio formats](/vertex-ai/generative-ai/docs/live-api#supported-audio-formats) (</vertex-ai/generative-ai/docs/live-api#supported-audio-formats>).

## System instructions

You can provide system instructions to better control the model's output and specify the tone and sentiment of audio responses.

System instructions are added to the prompt before the interaction begins and remain in effect for the entire session.

System instructions can only be set at the beginning of a session, immediately following the initial connection. To provide further input to the model during the session, use incremental content updates.

## Interruptions

Users can interrupt the model's output at any time. When Voice activity detection (VAD) detects an interruption, the ongoing generation is canceled and discarded. Only the information already sent to the client is retained in the session history. The server then sends a `BidiGenerateContentServerContent` message to report the interruption.

In addition, the Gemini server discards any pending function calls and sends a `BidiGenerateContentServerContent` message with the IDs of the canceled calls.

## Voices

To specify a voice, set the `voiceName` within the `speechConfig` object, as part of your [session configuration](#) (`#sessions`).

See the following JSON representation of a `speechConfig` object:

```
{
  "voiceConfig": {
    "prebuiltVoiceConfig": {
      "voiceName": " VOICE_NAME "
    }
  }
}
```

To see the list of supported voices, see [Change voice and language settings](#) (/vertex-ai/generative-ai/docs/live-api#voice-settings).

## Limitations

Consider the following limitations of Live API and Gemini 2.0 when you plan your project.

### Client authentication

Live API only provides server to server authentication and isn't recommended for direct client use. Client input should be routed through an intermediate application server for secure authentication with the Live API.

### Maximum session duration

The default maximum length of a conversation session is 10 minutes. For more information, see [Session length](#) (/vertex-ai/generative-ai/docs/live-api#session\_length).

### Voice activity detection (VAD)

By default, the model automatically performs voice activity detection (VAD) on a continuous audio input stream. VAD can be configured with the

**RealtimeInputConfig AutomaticActivityDetection**

(#google.cloud.aiplatform.v1beta1.RealtimeInputConfig.FIELDS.google.cloud.aiplatform.v1beta1.RealtimeInputConfig AutomaticActivityDetection.google.cloud.aiplatform.v1beta1.RealtimeInputConfig.automatic\_activity\_detection)

field of the [setup message](#) (#bidigeneratecontentsetup).

When the audio stream is paused for more than a second (for example, when the user switches off the microphone), an `AudioStreamEnd` event is sent to flush any cached audio. The client can resume sending audio data at any time.

Alternatively, the automatic VAD can be turned off by setting `RealtimeInputConfig.AutomaticActivityDetection.disabled` to `true` in the setup message. In this configuration the client is responsible for detecting user speech and sending `ActivityStart` (`#google.cloud.aiplatform.v1beta1.BidiGenerateContentRealtimeInput.FIELDS.google.cloud.aiplatform.v1beta1.BidiGenerateContentRealtimeInput.ActivityStart.google.cloud.aiplatform.v1beta1.BidiGenerateContentRealtimeInput.activity_start`)

and `ActivityEnd`

(`#google.cloud.aiplatform.v1beta1.BidiGenerateContentRealtimeInput.FIELDS.google.cloud.aiplatform.v1beta1.BidiGenerateContentRealtimeInput.ActivityEnd.google.cloud.aiplatform.v1beta1.BidiGenerateContentRealtimeInput.activity_end`)

messages at the appropriate times. An `AudioStreamEnd` isn't sent in this configuration. Instead, any interruption of the stream is marked by an `ActivityEnd` message.

## Additional limitations

Manual endpointing isn't supported.

Audio inputs and audio outputs negatively impact the model's ability to use function calling.

## Token count

Token count isn't supported.

## Rate limits

The following rate limits apply:

- 5,000 concurrent sessions per API key
- 4M tokens per minute

## Messages and events

### BidiGenerateContentClientContent

Incremental update of the current conversation delivered from the client. All the content here is unconditionally appended to the conversation history and used as part of the prompt to the model to generate content.

A message here will interrupt any current model generation.

Fields

turns[ ]	<p><b><u>Content</u></b></p> <p>(/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.Content)</p> <p>Optional. The content appended to the current conversation with the model.</p> <p>For single-turn queries, this is a single instance. For multi-turn queries, this is a repeated field that contains conversation history and latest request.</p>
turn_complete	<p><b>bool</b></p> <p>Optional. If true, indicates that the server content generation should start with the currently accumulated prompt. Otherwise, the server will await additional messages before starting generation.</p>

BidiGenerateContentRealtimeInput

User input that is sent in real time.

This is different from `ClientContentUpdate` in a few ways:

- Can be sent continuously without interruption to model generation.
- If there is a need to mix data interleaved across the `ClientContentUpdate` and the `RealtimeUpdate`, server attempts to optimize for best response, but there are no guarantees.
- End of turn is not explicitly specified, but is rather derived from user activity (for example, end of speech).
- Even before the end of turn, the data is processed incrementally to optimize for a fast start of the response from the model.
- Is always assumed to be the user's input (cannot be used to populate conversation history).

Fields

<b>media_chunks[ ]</b>	<p><b><u>Blob</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.Blob)</p> <p>Optional. Inlined bytes data for media input.</p>
<b>activity_start</b>	<p><b><u>ActivityStart</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.BidiGenerateContentRealtimeInput.ActivityStart)</p> <p>Optional. Marks the start of user activity. This can only be sent if automatic (i.e. server-side) activity detection is disabled.</p>
<b>activity_end</b>	<p><b><u>ActivityEnd</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.BidiGenerateContentRealtimeInput.ActivityEnd)</p> <p>Optional. Marks the end of user activity. This can only be sent if automatic (i.e. server-side) activity detection is disabled.</p>

ActivityEnd

This type has no fields.

Marks the end of user activity.

ActivityStart

This type has no fields.

Only one of the fields in this message must be set at a time. Marks the start of user activity.

BidiGenerateContentServerContent

Incremental server update generated by the model in response to client messages.

Content is generated as quickly as possible, and not in realtime. Clients may choose to buffer and play it out in realtime.

---

**Fields**

<b>turn_complete</b>	<b>bool</b>
	Output only. If true, indicates that the model is done generating. Generation will only start in response to additional client messages. Can be set alongside <b>content</b> , indicating that the <b>content</b> is the last in the turn.
<b>interrupted</b>	<b>bool</b>
	Output only. If true, indicates that a client message has interrupted current model generation. If the client is playing out the content in realtime, this is a good signal to stop and empty the current queue. If the client is playing out the content in realtime, this is a good signal to stop and empty the current playback queue.
<b>generation_complete</b>	<b>bool</b>
	Output only. If true, indicates that the model is done generating.
	When model is interrupted while generating there will be no 'generation_complete' message in interrupted turn, it will go through 'interrupted > turn_complete'.
	When model assumes realtime playback there will be delay between generation_complete and turn_complete that is caused by model waiting for playback to finish.
<b>grounding_metadata</b>	<b><u>GroundingMetadata</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.GroundingMetadata)
	Output only. Metadata specifies sources used to ground generated content.
<b>input_transcription</b>	<b><u>Transcription</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.BidiGenerateContentServerContent.Transcription)
	Optional. Input transcription. The transcription is independent to the model turn which means it doesn't imply any ordering between transcription and model turn.
<b>output_transcription</b>	<b><u>Transcription</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.BidiGenerateContentServerContent.Transcription)
	Optional. Output transcription. The transcription is independent to the model turn which means it doesn't imply any ordering between transcription and model turn.

---

Fields

model_turn	<p><b><u>Content</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.Content)</p> <p>Output only. The content that the model has generated as part of the current conversation with the user.</p>
------------	--

Transcription

Audio transcription message.

Fields

text	<p><b>string</b></p> <p>Optional. Transcription text.</p>
finished	<p><b>bool</b></p> <p>Optional. The bool indicates the end of the transcription.</p>

BidiGenerateContentSetup

Message to be sent in the first and only first client message. Contains configuration that will apply for the duration of the streaming session.

Clients should wait for a **BidiGenerateContentSetupComplete** message before sending any additional messages.

Fields

model	<p><b>string</b></p> <p>Required. The fully qualified name of the publisher model.</p> <p>Publisher model format: <b>projects/{project}/locations/{location}/publishers/*/models/*</b></p>
-------	--



Fields

generation_config	<p><b><u>GenerationConfig</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.GenerationConfig)</p> <p>Optional. Generation config.</p> <p>The following fields aren't supported:</p> <ul style="list-style-type: none"><li>• response_logprobs</li><li>• response_mime_type</li><li>• logprobs</li><li>• response_schema</li><li>• stop_sequence</li><li>• routing_config</li><li>• audio_timestamp</li></ul>
system_instruction	<p><b><u>Content</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.Content)</p> <p>Optional. The user provided system instructions for the model. Note: only text should be used in parts and content in each part will be in a separate paragraph.</p>
tools[]	<p><b><u>Tool</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.Tool)</p> <p>Optional. A list of <b>Tools</b> the model may use to generate the next response.</p> <p>A <b>Tool</b> is a piece of code that enables the system to interact with external systems to perform an action, or set of actions, outside of knowledge and scope of the model.</p>
session_resumption	<p><b><u>SessionResumptionConfig</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.SessionResumptionConfig)</p> <p>Optional. Configures session resumption mechanism. If included, the server will send periodical <b>SessionResumptionUpdate</b> messages to the client.</p>

## Fields

<b>context_window_compression</b>	<b><u>ContextWindowCompressionConfig</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.ContextWindowCompressionConfig)  Optional. Configures context window compression mechanism.  If included, server will compress context window to fit into given length.
<b>realtime_input_config</b>	<b><u>RealtimeInputConfig</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.RealtimeInputConfig)  Optional. Configures the handling of realtime input.
<b>input_audio_transcription</b>	<b><u>AudioTranscriptionConfig</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.BidiGenerateContentSetup.AudioTranscriptionConfig)  Optional. The transcription of the input aligns with the input audio language.
<b>output_audio_transcription</b>	<b><u>AudioTranscriptionConfig</u></b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.BidiGenerateContentSetup.AudioTranscriptionConfig)  Optional. The transcription of the output aligns with the language code specified for the output audio.

## AudioTranscriptionConfig

This type has no fields.

The audio transcription configuration.

## BidiGenerateContentSetupComplete

This type has no fields.

Sent in response to a **BidiGenerateContentSetup** message from the client.

## BidiGenerateContentToolCall

Request for the client to execute the `function_calls` and return the responses with the matching `ids`.

### Fields

<code>function_calls[ ]</code>	<div><div><u><b>FunctionCall</b></u></div><div>(/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.FunctionCall)</div></div> <div>Output only. The function call to be executed.</div>
--------------------------------	---

## BidiGenerateContentToolCallCancellation

Notification for the client that a previously issued `ToolCallMessage` with the specified `ids` should have been not executed and should be cancelled. If there were side-effects to those tool calls, clients may attempt to undo the tool calls. This message occurs only in cases where the clients interrupt server turns.

### Fields

<code>ids[ ]</code>	<div><div><b>string</b></div><div>Output only. The ids of the tool calls to be cancelled.</div></div>
---------------------	---

## BidiGenerateContentToolResponse

Client generated response to a `ToolCall` received from the server. Individual `FunctionResponse` objects are matched to the respective `FunctionCall` objects by the `id` field.

Note that in the unary and server-streaming `GenerateContent` APIs function calling happens by exchanging the `Content` parts, while in the bidi `GenerateContent` APIs function calling happens over these dedicated set of messages.

### Fields

<code>function_responses[ ]</code>	<div><div><u><b>FunctionResponse</b></u></div><div>(/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.FunctionResponse)</div></div>
------------------------------------	---

Fields

Optional. The response to the function calls.

RealtimeInputConfig

Configures the realtime input behavior in `BidiGenerateContent`.

Fields

<code>automatic_activity_detection</code>	<p><u><code>AutomaticActivityDetection</code></u> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.RealtimeInputConfig.AutomaticActivityDetection)</p> <p>Optional. If not set, automatic activity detection is enabled by default. If automatic voice detection is disabled, the client must send activity signals.</p>
<code>activity_handling</code>	<p><u><code>ActivityHandling</code></u> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.RealtimeInputConfig.ActivityHandling)</p> <p>Optional. Defines what effect activity has.</p>
<code>turn_coverage</code>	<p><u><code>TurnCoverage</code></u> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.RealtimeInputConfig.TurnCoverage)</p> <p>Optional. Defines which input is included in the user's turn.</p>

ActivityHandling

The different ways of handling user activity.

Enums

`ACTIVITY_HANDLING_UNSPECIFIED` If unspecified, the default behavior is `START_OF_ACTIVITY_INTERRUPTS`.

Enums

<b>START_OF_ACTIVITY_INTERRUPTS</b>	If true, start of activity will interrupt the model's response (also called "barge in"). The model's current response will be cut-off in the moment of the interruption. This is the default behavior.
<b>NO_INTERRUPTION</b>	The model's response will not be interrupted.

# AutomaticActivityDetection

Configures automatic detection of activity.

Fields

<b>start_of_speech_sensitivity</b>	<b>StartSensitivity</b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.RealtimeInputConfig.AutomaticActivityDetection.StartSensitivity)  Optional. Determines how likely speech is to be detected.
<b>end_of_speech_sensitivity</b>	<b>EndSensitivity</b> (/vertex-ai/generative-ai/docs/reference/rpc/google.cloud.aiplatform.v1beta1#google.cloud.aiplatform.v1beta1.RealtimeInputConfig.AutomaticActivityDetection.EndSensitivity)  Optional. Determines how likely detected speech is ended.
<b>prefix_padding_ms</b>	<b>int32</b>  Optional. The required duration of detected speech before start-of-speech is committed. The lower this value the more sensitive the start-of-speech detection is and the shorter speech can be recognized. However, this also increases the probability of false positives.
<b>silence_duration_ms</b>	<b>int32</b>  Optional. The required duration of detected silence (or non-speech) before end-of-speech is committed. The larger this value, the longer speech gaps can be without interrupting the user's activity but this will increase the model's latency.
<b>disabled</b>	<b>bool</b>  Optional. If enabled, detected voice and text input count as activity. If disabled, the client must send activity signals.

# EndSensitivity

End of speech sensitivity.

Enums

END\_SENSITIVITY\_UNSPECIFIED

The default is END\_SENSITIVITY\_LOW.

END\_SENSITIVITY\_HIGH

Automatic detection ends speech more often.

END\_SENSITIVITY\_LOW

Automatic detection ends speech less often.

# StartSensitivity

Start of speech sensitivity.

Enums

START\_SENSITIVITY\_UNSPECIFIED

The default is START\_SENSITIVITY\_LOW.

START\_SENSITIVITY\_HIGH

Automatic detection will detect the start of speech more often.

START\_SENSITIVITY\_LOW

Automatic detection will detect the start of speech less often.

# TurnCoverage

Options about which input is included in the user's turn.

Enums

TURN\_COVERAGE\_UNSPECIFIED

If unspecified, the default behavior is TURN\_INCLUDES\_ALL\_INPUT.

TURN\_INCLUDES\_ONLY\_ACTIVITY

The users turn only includes activity since the last turn, excluding inactivity (e.g. silence on the audio stream).

Enums

**TURN\_INCLUDES\_ALL\_INP**UTThe users turn includes all realtime input since the last turn, including inactivity (e.g. silence on the audio stream). This is the default behavior.

UsageMetadata

Metadata on the usage of the cached content.

Fields

<b>total_token_count</b>	<b>int32</b>	Total number of tokens that the cached content consumes.
<b>text_count</b>	<b>int32</b>	Number of text characters.
<b>image_count</b>	<b>int32</b>	Number of images.
<b>video_duration_seconds</b>	<b>int32</b>	Duration of video in seconds.
<b>audio_duration_seconds</b>	<b>int32</b>	Duration of audio in seconds.

GoAway

Server will not be able to service client soon.

Fields

<b>time_left</b>	<b><u>Duration</u></b> ( <a href="https://protobuf.dev/reference/protobuf/google.protobuf/#duration">https://protobuf.dev/reference/protobuf/google.protobuf/#duration</a> )
	The remaining time before the connection will be terminated as ABORTED. The minimal time returned here is specified differently together with the rate limits for a given model.

SessionResumptionUpdate

Update of the session resumption state.

Only sent if `BidiGenerateContentSetup.session_resumption` was set.

Fields

<code>new_handle</code>	<code>string</code>	New handle that represents state that can be resumed. Empty if <code>resumable=false</code> .
<code>resumable</code>	<code>bool</code>	<p>True if session can be resumed at this point.</p> <p>It might be not possible to resume session at some points. In that case we send update empty <code>new_handle</code> and <code>resumable=false</code>. Example of such case could be model executing function calls or just generating. Resuming session (using previous session token) in such state will result in some data loss.</p>
<code>last_consumed_client_int64_message_index</code>		<p>Index of last message sent by client that is included in state represented by this <code>SessionResumptionToken</code>. Only sent when <code>SessionResumptionConfig.transparent</code> is set.</p> <p>Presence of this index allows users to transparently reconnect and avoid issue of losing some part of realtime audio input/video. If client wishes to temporarily disconnect (for example as result of receiving <code>GoAway</code>) they can do it without losing state by buffering messages sent since last <code>SessionResumptionTokenUpdate</code>. This field will enable them to limit buffering (avoid keeping all requests in RAM).</p> <p>It will not be used for 'resumption to restore state' some time later -- in those cases partial audio and video frames are likely not needed.</p>

What's next

- Learn more about [function calling](/vertex-ai/generative-ai/docs/multimodal/function-calling) (/vertex-ai/generative-ai/docs/multimodal/function-calling).
- See the [Function calling reference](/vertex-ai/generative-ai/docs/model-reference/function-calling) (/vertex-ai/generative-ai/docs/model-reference/function-calling) for examples.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (https://developers.google.com/site-policies). Java is a registered trademark of Oracle and/or its affiliates.



Last updated 2025-06-06 UTC.