

Generate content with the Gemini API in Vertex AI

Use `generateContent` or `streamGenerateContent` to generate content with Gemini.

The Gemini model family includes models that work with multimodal prompt requests. The term multimodal indicates that you can use more than one modality, or type of input, in a prompt. Models that aren't multimodal accept prompts only with text. Modalities can include text, audio, video, and more.

Create a Google Cloud account to get started

To start using the Gemini API in Vertex AI, [create a Google Cloud account](#)

(<https://console.cloud.google.com/freetrial?redirectPath=/marketplace/product/google/cloudai companion.googleapis.com>).

After creating your account, use this document to review the Gemini model [request body](#) (#request), [model parameters](#) (#parameters), [response body](#) (#response), and some sample [requests](#) (#sample-requests).

When you're ready, see the [Gemini API in Vertex AI quickstart](#) (/vertex-ai/generative-ai/docs/start/quickstarts/quickstart-multimodal) to learn how to send a request to the Gemini API in Vertex AI using a programming language SDK or the REST API.

Supported models

All Gemini models support content generation.

Note: Adding a lot of images to a request increases response latency.

Parameter list

See [examples](#) (#sample-requests) for implementation details.

Request body

```
{  
  "cachedContent": string,  
  "contents": [  
    {  
      "role": string,  
      "parts": [  
        {  
          // Union field data can be only one of the following:  
          "text": string,  
          "inlineData": {  
            "mimeType": string,  
            "data": string  
          },  
          "fileData": {  
            "mimeType": string,  
            "fileUri": string  
          },  
          // End of list of possible types for union field data.  
  
          "videoMetadata": {  
            "startOffset": {  
              "seconds": integer,  
              "nanos": integer  
            },  
            ...  
          },  
          ...  
        }  
      ]  
    }  
  ]  
}
```

```
"endOffset": {
    "seconds": integer,
    "nanos": integer
},
"fps": double
}
]
}
],
"systemInstruction": {
    "role": string,
    "parts": [
        {
            "text": string
        }
    ]
},
"tools": [
{
    "functionDeclarations": [
        {
            "name": string,
            "description": string,
            "parameters": {
                object (https://spec.openapis.org/oas/v3.0.3#schema)nAPI Object Schema (https://spec.openapis.org/oas/v3.0.3#sch)
            }
        }
    ]
},
"safetySettings": [
{
```

```
        "category": enum (HarmCategory),
        "threshold": enum (HarmBlockThreshold)
    },
],
"generationConfig": {
    "temperature": number,
    "topP": number,
    "topK": number,
    "candidateCount": integer,
    "maxOutputTokens": integer,
    "presencePenalty": float,
    "frequencyPenalty": float,
    "stopSequences": [
        string
    ],
    "responseMimeType": string,
    "responseSchema": schema \(/vertex-ai/docs/reference/rest/v1/projects.locations.cachedContents#Schema\),
    "seed": integer,
    "responseLogprobs": boolean,
    "logprobs": integer,
    "audioTimestamp": boolean,
    "thinkingConfig": {
        "thinkingBudget": integer
    }
},
"labels": {
    string: string
}
}
```

The request body contains data with the following parameters:

Parameters

cachedContent	Optional: string The name of the cached content used as context to serve the prediction. Format: projects/{project}/locations/{location}/cachedContents/{cachedContent}
contents	Required: Content The content of the current conversation with the model. For single-turn queries, this is a single instance. For multi-turn queries, this is a repeated field that contains conversation history and the latest request.
systemInstruction	Optional: Content Available for gemini-2.0-flash and gemini-2.0-flash-lite . Instructions for the model to steer it toward better performance. For example, "Answer as concisely as possible" or "Don't use technical terms in your response". The text strings count toward the token limit. The role field of systemInstruction is ignored and doesn't affect the performance of the model.
<p>★ Note: Only text should be used in parts and content in each part should be in a separate paragraph.</p>	
tools	Optional. 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. See Function calling (/vertex-ai/generative-ai/docs/model-reference/function-calling).
toolConfig	Optional. See Function calling (/vertex-ai/generative-ai/docs/model-reference/function-calling).

safetySettings	Optional: SafetySetting Per request settings for blocking unsafe content. Enforced on GenerateContentResponse.candidates .
generationConfig	Optional: GenerationConfig Generation configuration settings.
labels	Optional: string Metadata that you can add to the API call in the format of key-value pairs.

contents

The base structured data type containing multi-part content of a message.

This class consists of two main properties: **role** and **parts**. The **role** property denotes the individual producing the content, while the **parts** property contains multiple elements, each representing a segment of data within a message.

Parameters

role	string The identity of the entity that creates the message. The following values are supported: <ul style="list-style-type: none">• user: This indicates that the message is sent by a real person, typically a user-generated message.• model: This indicates that the message is generated by the model. The model value is used to insert messages from the model into the conversation during multi-turn conversations.
-------------	---

parts**Part**

A list of ordered parts that make up a single message. Different parts may have different [IANA MIME types](https://www.iana.org/assignments/media-types/media-types.xml) (<https://www.iana.org/assignments/media-types/media-types.xml>).

For limits on the inputs, such as the maximum number of tokens or the number of images, see the model specifications on the [Google models](#) (/vertex-ai/generative-ai/docs/learn/models) page.

To compute the number of tokens in your request, see [Get token count](#) (/vertex-ai/generative-ai/docs/multimodal/get-token-count).

parts

A data type containing media that is part of a multi-part **Content** message.

Parameters

text	Optional: string
-------------	-------------------------

A text prompt or code snippet.

inlineData	Optional: Blob
-------------------	-----------------------

Inline data in raw bytes.

For **gemini-2.0-flash-lite** and **gemini-2.0-flash**, you can specify up to 3000 images by using **inlineData**.

fileData	Optional: fileData
-----------------	---------------------------

Data stored in a file.

functionCall	Optional: FunctionCall .
---------------------	---------------------------------

It contains a string representing the `FunctionDeclaration.name` field and a structured JSON object containing any parameters for the function call predicted by the model.

See [Function calling](#) (/vertex-ai/generative-ai/docs/model-reference/function-calling).

functionResponse

Optional: `FunctionResponse`.

The result output of a `FunctionCall` that contains a string representing the `FunctionDeclaration.name` field and a structured JSON object containing any output from the function call. It is used as context to the model.

See [Function calling](#) (/vertex-ai/generative-ai/docs/model-reference/function-calling).

videoMetadata

Optional: `VideoMetadata`

For video input, the start and end offset of the video in `Duration` (<https://protobuf.dev/reference/protobuf/google.protobuf/#duration>) format, and the frame rate of the video . For example, to specify a 10 second clip starting at 1:00 with a frame rate of 10 frames per second, set the following:

- "startOffset": { "seconds": 60 }
- "endOffset": { "seconds": 70 }
- "fps": 10.0

The metadata should only be specified while the video data is presented in `inlineData` or `fileData`.

blob

Content blob. If possible send as text rather than raw bytes.

Parameters

mimeType

`string`

The media type of the file specified in the `data` or `fileUri` fields. Acceptable values include the following:

 Click to expand MIME types

- `application/pdf`
- `audio/mpeg`
- `audio/mp3`
- `audio/wav`
- `image/png`
- `image/jpeg`
- `image/webp`
- `text/plain`
- `video/mov`
- `video/mpeg`
- `video/mp4`
- `video/mpg`
- `video/avi`
- `video/wmv`
- `video/mpegps`
- `video/flv`

For `gemini-2.0-flash-lite` and `gemini-2.0-flash`, the maximum length of an audio file is 8.4 hours and the maximum length of a video file (without audio) is one hour. For more information, see [Gemini audio](#) (/vertex-ai/generative-ai/docs/multimodal/audio-understanding#audio-requirements) and [video](#) (/vertex-ai/generative-ai/docs/multimodal/video-understanding#video-requirements).

Text files must be UTF-8 encoded. The contents of the text file count toward the token limit.

There is no limit on image resolution.

data	bytes
	The base64 encoding (/vertex-ai/generative-ai/docs/image/base64-encode) of the image, PDF, or video to include inline in the prompt. When including media inline, you must also specify the media type (mimeType) of the data.
	Size limit: 20MB

FileData

URI or web-URL data.

Parameters

mimeType	string
	IANA MIME type (https://www.iana.org/assignments/media-types/media-types.xml) of the data.
fileUri	string

The URI or URL of the file to include in the prompt. Acceptable values include the following:

- **Cloud Storage bucket URI:** The object must either be publicly readable or reside in the same Google Cloud project that's sending the request. For **gemini-2.0-flash** and **gemini-2.0-flash-lite**, the size limit is 2 GB.
- **HTTP URL:** The file URL must be publicly readable. You can specify one video file, one audio file, and up to 10 image files per request. Audio files, video files, and documents can't exceed 15 MB.
- **YouTube video URL:** The YouTube video must be either owned by the account that you used to sign in to the Google Cloud console or is public. Only one YouTube video URL is supported per request.

When specifying a **fileURI**, you must also specify the media type (**mimeType**) of the file. If VPC Service Controls is enabled, specifying a media file URL for **fileURI** is not supported.

functionCall

A predicted **functionCall** returned from the model that contains a string representing the **functionDeclaration.name** and a structured JSON object containing the parameters and their values.

Parameters

name	string
-------------	---------------

The name of the function to call.

args	Struct
-------------	---------------

The function parameters and values in JSON object format.

See [Function calling](#) (/vertex-ai/generative-ai/docs/model-reference/function-calling) for parameter details.

functionResponse

The resulting output from a **FunctionCall** that contains a string representing the **FunctionDeclaration.name**. Also contains a structured JSON object with the output from the function (and uses it as context for the model). This should contain the result of a **FunctionCall** made based on model prediction.

Parameters

name	string
	The name of the function to call.
response	Struct
	The function response in JSON object format.

videoMetadata

Metadata describing the input video content.

Parameters

startOffset	Optional: <code>google.protobuf.Duration</code>
	The start offset of the video.
endOffset	Optional: <code>google.protobuf.Duration</code>
	The end offset of the video.
fps	Optional: <code>double</code>
	The frame rate of the video sent to the model. Defaults to <code>1.0</code> if not specified. The minimum accepted value is as low as, but not including, <code>0.0</code> . The maximum value is <code>24.0</code> .

safetySetting

Safety settings.

Parameters

category	Optional: <code>HarmCategory</code>
-----------------	-------------------------------------

The safety category to configure a threshold for. Acceptable values include the following:

[Click to expand safety categories](#)

- `HARM_CATEGORY_SEXUALLY_EXPLICIT`
- `HARM_CATEGORY_HATE_SPEECH`
- `HARM_CATEGORY_HARASSMENT`
- `HARM_CATEGORY_DANGEROUS_CONTENT`

threshold	Optional: <code>HarmBlockThreshold</code>
------------------	---

The threshold for blocking responses that could belong to the specified safety category based on probability.

- `OFF`
- `BLOCK_NONE`
- `BLOCK_LOW_AND ABOVE`
- `BLOCK_MEDIUM_AND ABOVE`
- `BLOCK_ONLY_HIGH`

method	Optional: <code>HarmBlockMethod</code>
---------------	--

Specify if the threshold is used for probability or severity score. If not specified, the threshold is used for probability score.

harmCategory

Harm categories that block content.

Parameters

HARM_CATEGORY_UNSPECIFIED	The harm category is unspecified.
HARM_CATEGORY_HATE_SPEECH	The harm category is hate speech.
HARM_CATEGORY_DANGEROUS_CONTENT	The harm category is dangerous content.
HARM_CATEGORY_HARASSMENT	The harm category is harassment.
HARM_CATEGORY_SEXUALLY_EXPLICIT	The harm category is sexually explicit content.

harmBlockThreshold

Probability thresholds levels used to block a response.

Parameters

HARM_BLOCK_THRESHOLD_UNSPECIFIED	Unspecified harm block threshold.
BLOCK_LOW_AND ABOVE	Block low threshold and higher (i.e. block more).
BLOCK_MEDIUM_AND ABOVE	Block medium threshold and higher.
BLOCK_ONLY_HIGH	Block only high threshold (i.e. block less).
BLOCK_NONE	Block none.

OFF	Switches off safety if all categories are turned OFF
-----	--

harmBlockMethod

A probability threshold that blocks a response based on a combination of probability and severity.

Parameters

HARM_BLOCK_METHOD_UNSPECIFIED	The harm block method is unspecified.
-------------------------------	---------------------------------------

SEVERITY	The harm block method uses both probability and severity scores.
----------	--

PROBABILITY	The harm block method uses the probability score.
-------------	---

generationConfig

Configuration settings used when generating the prompt.

Parameters

temperature	Optional: float
-------------	-----------------

The temperature is used for sampling during response generation, which occurs when `topP` and `topK` are applied. Temperature controls the degree of randomness in token selection. Lower temperatures are good for prompts that require a less open-ended or creative response, while higher temperatures can lead to more diverse or creative results. A temperature of 0 means that the highest probability tokens are always selected. In this case, responses for a given prompt are mostly deterministic, but a small amount of variation is still possible.

If the model returns a response that's too generic, too short, or the model gives a fallback response, try increasing the temperature. If the model enters infinite generation, increasing the temperature to at least 0.1 may lead to

improved results.

1.0 is the recommended starting value for temperature.

- Range for **gemini-2.0-flash-lite**: **0.0 - 2.0** (default: **1.0**)
- Range for **gemini-2.0-flash**: **0.0 - 2.0** (default: **1.0**)

For more information, see [Content generation parameters](#) (/vertex-ai/generative-ai/docs/multimodal/content-generation-parameters#temperature).

topP

Optional: **float**

If specified, nucleus sampling is used.

[Top-P](#) (/vertex-ai/generative-ai/docs/multimodal/content-generation-parameters#top-p) changes how the model selects tokens for output. Tokens are selected from the most (see top-K) to least probable until the sum of their probabilities equals the top-P value. For example, if tokens A, B, and C have a probability of 0.3, 0.2, and 0.1 and the top-P value is **0.5**, then the model will select either A or B as the next token by using temperature and excludes C as a candidate.

Specify a lower value for less random responses and a higher value for more random responses.

- Range: **0.0 - 1.0**
 - Default for **gemini-2.0-flash-lite**: **0.95**
 - Default for **gemini-2.0-flash**: **0.95**
-

candidateCount

Optional: **int**

The number of response variations to return. For each request, you're charged for the output tokens of all candidates, but are only charged once for the input tokens.

Specifying multiple candidates is a Preview feature that works with **generateContent** (**streamGenerateContent** is not supported). The following models are supported:

- **gemini-2.0-flash-lite**: 1-8, default: 1
- **gemini-2.0-flash**: 1-8, default: 1

maxOutputTokens

Optional: int

Maximum number of tokens that can be generated in the response. A token is approximately four characters. 100 tokens correspond to roughly 60-80 words.

Specify a lower value for shorter responses and a higher value for potentially longer responses.

For more information, see [Content generation parameters](#) (/vertex-ai/generative-ai/docs/multimodal/content-generation-parameters#max-output-tokens).

stopSequences

Optional: `List[string]`

Specifies a list of strings that tells the model to stop generating text if one of the strings is encountered in the response. If a string appears multiple times in the response, then the response truncates where it's first encountered. The strings are case-sensitive.

For example, if the following is the returned response when `stopSequences` isn't specified:

```
public static string reverse(string myString)
```

Then the returned response with `stopSequences` set to `["Str", "reverse"]` is:

```
public static string
```

Maximum 5 items in the list.

For more information, see [Content generation parameters](#) (/vertex-ai/generative-ai/docs/multimodal/content-generation-parameters#stop-sequences).

presencePenalty

Optional: float

Positive penalties.

Positive values penalize tokens that already appear in the generated text, increasing the probability of generating more diverse content.

The maximum value for `presencePenalty` is up to, but not including, `2.0`. Its minimum value is `-2.0`.

frequencyPenalty

Optional: `float`

Positive values penalize tokens that repeatedly appear in the generated text, decreasing the probability of repeating content.

This maximum value for `frequencyPenalty` is up to, but not including, `2.0`. Its minimum value is `-2.0`.

responseMimeType

Optional: `string (enum)`

The output response MIME type of the generated candidate text.

The following MIME types are supported:

- `application/json`: JSON response in the candidates.
- `text/plain` (default): Plain text output.
- `text/x.enum`: For classification tasks, output an enum value as defined in the response schema.

Specify the appropriate response type to avoid unintended behaviors. For example, if you require a JSON-formatted response, specify `application/json` and not `text/plain`.

`text/plain` isn't supported for use with `responseSchema`.

responseSchema

Optional: [schema \(/vertex-ai/docs/reference/rest/v1/projects.locations.cachedContents#Schema\)](#)

The schema that generated candidate text must follow. For more information, see [Control generated output](#) (/vertex-ai/generative-ai/docs/multimodal/control-generated-output).

To use this parameter, you must specify a supported mime type other than `text/plain` for the `responseMimeType` parameter.

seed	Optional: <code>int</code>
	When seed is fixed to a specific value, the model makes a best effort to provide the same response for repeated requests. Deterministic output isn't guaranteed. Also, changing the model or parameter settings, such as the temperature, can cause variations in the response even when you use the same seed value. By default, a random seed value is used.
responseLogprobs	Optional: <code>boolean</code>
	If true, returns the log probabilities of the tokens that were chosen by the model at each step. By default, this parameter is set to <code>false</code> .
logprobs	Optional: <code>int</code>
	Returns the log probabilities of the top candidate tokens at each generation step. The model's chosen token might not be the same as the top candidate token at each step. Specify the number of candidates to return by using an integer value in the range of 1-20. You must enable <u>responseLogprobs</u> (#responseLogprobs) to use this parameter.
audioTimestamp	Optional: <code>boolean</code>
	Available for the following models: <ul style="list-style-type: none">• Gemini 2.0 Flash-Lite• Gemini 2.0 Flash Enables timestamp understanding for audio-only files. This is a preview feature.
thinkingConfig	Optional: <code>object</code>

Configuration for the model's [thinking process](#) (/vertex-ai/generative-ai/docs/thinking) for Gemini 2.5 models.

The `thinkingConfig` object contains the following field:

- **thinkingBudget: integer**. By default, the model automatically controls how much it thinks up to a maximum of 8,192 tokens.

Response body

```
{  
  "candidates": [  
    {  
      "content": {  
        "parts": [  
          {  
            "text": string  
          }  
        ]  
      },  
      "finishReason": enum (FinishReason),  
      "safetyRatings": [  
        {  
          "category": enum (HarmCategory),  
          "probability": enum (HarmProbability),  
          "blocked": boolean  
        }  
      ],  
      "citationMetadata": {  
        "citations": [  
          {  
            "startIndex": integer,  
            "endIndex": integer  
          }  
        ]  
      }  
    }  
  ]  
}
```

```
"endIndex": integer,  
"uri": string,  
"title": string,  
"license": string,  
"publicationDate": {  
    "year": integer,  
    "month": integer,  
    "day": integer  
}  
}  
]  
},  
"avgLogprobs": double,  
"logprobsResult": {  
    "topCandidates": [  
        {  
            "candidates": [  
                {  
                    "token": string,  
                    "logProbability": float  
                }  
            ]  
        }  
    ],  
    "chosenCandidates": [  
        {  
            "token": string,  
            "logProbability": float  
        }  
    ]  
}  
},  
],
```

```
"usageMetadata": {  
    "promptTokenCount": integer,  
    "candidatesTokenCount": integer,  
    "totalTokenCount": integer  
},  
"modelVersion": string  
}
```

Response element Description

modelVersion The model and version used for generation. For example: **gemini-2.0-flash-lite-001**.

text The generated text.

finishReason The reason why the model stopped generating tokens. If empty, the model has not stopped generating the tokens. Because the response uses the prompt for context, it's not possible to change the behavior of how the model stops generating tokens.

- **FINISH_REASON_STOP**: Natural stop point of the model or provided stop sequence.
- **FINISH_REASON_MAX_TOKENS**: The maximum number of tokens as specified in the request was reached.
- **FINISH_REASON_SAFETY**: Token generation was stopped because the response was flagged for safety reasons. Note that **Candidate.content** is empty if content filters block the output.
- **FINISH_REASON_RECITATION**: The token generation was stopped because the response was flagged for unauthorized citations.
- **FINISH_REASON_BLOCKLIST**: Token generation was stopped because the response includes blocked terms.
- **FINISH_REASON_PROHIBITED_CONTENT**: Token generation was stopped because the response was flagged for prohibited content, such as child sexual abuse material (CSAM).
- **FINISH_REASON_SPII**: Token generation was stopped because the response was flagged for sensitive personally identifiable information (SPII).

- **FINISH_REASON_MALFORMED_FUNCTION_CALL**: Candidates were blocked because of malformed and unparsable function call.
- **FINISH_REASON_OTHER**: All other reasons that stopped the token
- **FINISH_REASON_UNSPECIFIED**: The finish reason is unspecified.

category	The safety category to configure a threshold for. Acceptable values include the following:  Click to expand safety categories <ul style="list-style-type: none">• HARM_CATEGORY_SEXUALLY_EXPLICIT• HARM_CATEGORY_HATE_SPEECH• HARM_CATEGORY_HARASSMENT• HARM_CATEGORY_DANGEROUS_CONTENT
probability	The harm probability levels in the content. <ul style="list-style-type: none">• HARM_PROBABILITY_UNSPECIFIED• NEGLIGIBLE• LOW• MEDIUM• HIGH
blocked	A boolean flag associated with a safety attribute that indicates if the model's input or output was blocked.
startIndex	An integer that specifies where a citation starts in the content . The startIndex is in bytes and calculated from the response encoded in UTF-8.
endIndex	An integer that specifies where a citation ends in the content . The endIndex is in bytes and calculated from the response encoded in UTF-8.

url	The URL of a citation source. Examples of a URL source might be a news website or a GitHub repository.
title	The title of a citation source. Examples of source titles might be that of a news article or a book.
license	The license associated with a citation.
publication	The date a citation was published. Its valid formats are YYYY, YYYY-MM, and YYYY-MM-DD.
Date	
avgLogprobs	Average log probability of the candidate.
logprobsResult	Returns the top candidate tokens (topCandidates) and the actual chosen tokens (chosenCandidates) at each step.
token	Generative AI models break down text data into tokens for processing, which can be characters, words, or phrases.
logProbability	A log probability value that indicates the model's confidence for a particular token.
promptTokenCount	Number of tokens in the request.
candidatesTokenCount	Number of tokens in the response(s).
totalTokenCount	Number of tokens in the request and response(s).

Examples

Text Generation

Generate a text response from a text input.

[GenAI SDK for Python](#)
[Python \(OpenAI\)...](#)[Go \(#go\)](#)

```
from google import genai
from google.genai.types import HttpOptions

client = genai.Client(http_options=HttpOptions(api_version="v1"))
response = client.models.generate_content(
    model="gemini-2.5-flash",
    contents="How does AI work?",
)
print(response.text)
# Example response:
# Okay, let's break down how AI works. It's a broad field, so I'll focus on the ...
#
# Here's a simplified overview:
# ...
```

Using multimodal prompt

Generate a text response from a multimodal input, such as text and an image.

[GenAI SDK for Python](#)
[Python \(OpenAI\)...](#)
(#gen-ai-sdk-for-python)[Go \(#go\)](#)

```
from google import genai
from google.genai.types import HttpOptions, Part

client = genai.Client(http_options=HttpOptions(api_version="v1"))
```

```
response = client.models.generate_content(
    model="gemini-2.5-flash",
    contents=[
        "What is shown in this image?",
        Part.from_uri(
            file_uri="gs://cloud-samples-data/generative-ai/image/scones.jpg",
            mime_type="image/jpeg",
        ),
    ],
)
print(response.text)
# Example response:
# The image shows a flat lay of blueberry scones arranged on parchment paper. There are ...
```

Streaming text response

Generate a streaming model response from a text input.

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

[Go \(#go\)](#)

```
from google import genai
from google.genai.types import HttpOptions

client = genai.Client(http_options=HttpOptions(api_version="v1"))

for chunk in client.models.generate_content_stream(
    model="gemini-2.5-flash",
    contents="Why is the sky blue?",
):
    print(chunk)
```

```
    print(chunk.text, end="")
# Example response:
# The
# sky appears blue due to a phenomenon called **Rayleigh scattering**. Here's
# a breakdown of why:
# ...
```

Model versions

To use the [auto-updated version](#) (/vertex-ai/generative-ai/docs/learn/model-versioning#auto-updated-version), specify the model name without the trailing version number, for example `gemini-2.0-flash` instead of `gemini-2.0-flash-001`.

For more information, see [Gemini model versions and lifecycle](#) (/vertex-ai/generative-ai/docs/learn/model-versioning#gemini-model-versions).

What's next

- Learn more about the [Gemini API in Vertex AI](#) (/vertex-ai/generative-ai/docs/model-reference/gemini).
- Learn more about [Function calling](#) (/vertex-ai/generative-ai/docs/multimodal/function-calling).
- Learn more about [Grounding responses for Gemini models](#) (/vertex-ai/generative-ai/docs/multimodal/ground-gemini).

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/>), and code samples are licensed under the [Apache 2.0 License](#) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](#) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2025-10-24 UTC.