

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).



# Use the Count Tokens API

This page shows you how to get the token count and the number of billable characters for a prompt by using the `countTokens` API.

**Important:** Instead of using the `countTokens` API, we recommend that you use integrated tokenizer of the Vertex AI SDK for Python for getting token count. For details, see [List and count tokens](#) (/vertex-ai/generative-ai/docs/multimodal/list-token).

## Supported models

The following multimodal models support getting an estimate of the prompt token count:

- [Gemini 2.0 Flash with image generation](#) (/vertex-ai/generative-ai/docs/models/gemini/2-0-flash) 
- [Vertex AI Model Optimizer](#) (/vertex-ai/generative-ai/docs/model-reference/vertex-ai-model-optimizer) 
- [Gemini 2.5 Pro](#) (/vertex-ai/generative-ai/docs/models/gemini/2-5-pro)
- [Gemini 2.5 Flash](#) (/vertex-ai/generative-ai/docs/models/gemini/2-5-flash)
- [Gemini 2.0 Flash](#) (/vertex-ai/generative-ai/docs/models/gemini/2-0-flash)
- [Gemini 2.0 Flash-Lite](#) (/vertex-ai/generative-ai/docs/models/gemini/2-0-flash-lite)

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

## Get the token count for a prompt

You can get the token count estimate and the number of billable characters for a prompt by using the Vertex AI API.

**Important:** The input format for **CountTokens** depends on the model you use. Each input format is the same as the **Predict** input format.

<a href="#">Console (#console)</a>	<a href="#">Python Gen AI SDK</a>	<a href="#">Go Gen AI SDK</a>	<a href="#">Node.js Gen AI SDK</a>	<a href="#">Java Gen AI SDK</a>	<a href="#">REST (#rest)</a>
<h2>Install</h2> <p><code>pip install --upgrade google-genai</code></p> <p>To learn more, see the <a href="https://googleapis.github.io/python-genai/">SDK reference documentation</a> (https://googleapis.github.io/python-genai/).</p> <p>Set environment variables to use the Gen AI SDK with Vertex AI:</p> <pre># Replace the `GOOGLE_CLOUD_PROJECT` and `GOOGLE_CLOUD_LOCATION` values # with appropriate values for your project. export GOOGLE_CLOUD_PROJECT=<u>GOOGLE_CLOUD_PROJECT</u> export GOOGLE_CLOUD_LOCATION=<u>global</u> export GOOGLE_GENAI_USE_VERTEXAI=True</pre> <pre>from google import genai from google.genai.types import HttpOptions  client = genai.Client(http_options=HttpOptions(api_version="v1")) response = client.models.count_tokens(     model="gemini-2.5-flash",     contents="What's the highest mountain in Africa?", ) print(response) # Example output: # total_tokens=10 # cached_content_token_count=None</pre>					

**Example for text with image or video:**

n-

Python   Go   Node.js   Java   REST (#rest)

## Install

```
pip install --upgrade google-genai
```

To learn more, see the [SDK reference documentation](https://googleapis.github.io/python-genai/) (https://googleapis.github.io/python-genai/).

Set environment variables to use the Gen AI SDK with Vertex AI:

```
# Replace the `GOOGLE_CLOUD_PROJECT` and `GOOGLE_CLOUD_LOCATION` values
# with appropriate values for your project.
export GOOGLE_CLOUD_PROJECT=GOOGLE_CLOUD_PROJECT
export GOOGLE_CLOUD_LOCATION=global
export GOOGLE_GENAI_USE_VERTEXAI=True

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

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

contents = [
    Part.from_uri(
        file_uri="gs://cloud-samples-data/generative-ai/video/pixel8.mp4",
        mime_type="video/mp4",
    ),
    "Provide a description of the video.",
]

response = client.models.count_tokens(
    model="gemini-2.5-flash",
    contents=contents,
)
print(response)
# Example output:
# total_tokens=16252 cached_content_token_count=None
```

## Pricing and quota

There is no charge or quota restriction for using the CountTokens API. The maximum quota for the CountTokens API is 3000 requests per minute.

## What's next

- Learn how to use use Vertex AI SDK for Python to [list and count tokens](#) (/blob/main/vertexai/preview/tokenization.p) ([Preview](#) (/products#product-launch-stages))
- Learn about sending chat prompts and [text generation](#) (/vertex-ai/generative-ai/docs/multimodal/send-chat-prompts-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/) (<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 2025-07-26 UTC.