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).

Text generation

To see an example of getting started with Chat with the Gemini Pro model, run the "Getting Started with Chat with the Gemini Pro model" Jupyter notebook in one of the following environments:

Open in Colab (https://colab.research.google.com/github/GoogleCloudPlatform/generative-ai/blob/main/gemini/getting-started/intro_gemini_chat.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%2Fgetting-started%2Fintro_gemini_chat.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%2Fgetting-started%2Fintro_gemini_chat.ipynb)
| View on GitHub (https://github.com/GoogleCloudPlatform/generative-ai/blob/main/gemini/getting-started/intro_gemini_chat.ipynb)

This page shows you how to send chat prompts to a Gemini model by using the Google Cloud console, REST API, and supported SDKs.

To learn how to add images and other media to your request, see Image understanding (/vertex-ai/generative-ai/docs/multimodal/image-understanding).

For a list of languages supported by Gemini, see Language support (/vertex-ai/generative-ai/docs/learn/models#languages-gemini).

To explore the generative AI models and APIs that are available on Vertex AI, go to Model Garden in the Google Cloud console.

Go to Model Garden (https://console.cloud.google.com/vertex-ai/model-garden)

If you're looking for a way to use Gemini directly from your mobile and web apps, see the Firebase AI Logic client SDKs (https://firebase.google.com/docs/ai-logic) for Swift, Android, Web, Flutter, and Unity apps.

Generate text

For testing and iterating on chat prompts, we recommend using the Google Cloud console. To send prompts programmatically to the model, you can use the REST API, Google Gen AI SDK, Vertex AI SDK for Python, or one of the other supported libraries and SDKs.

You can use system instructions to steer the behavior of the model based on a specific need or use case. For example, you can define a persona or role for a chatbot that responds to customer service requests. For more information, see the system instructions code samples (/vertex-ai/generative-ai/docs/learn/prompts/system-instructions#code_samples).

You can use the Google Gen AI SDK (/vertex-ai/generative-ai/docs/gemini-v2#google-gen) to send requests if you're using Gemini 2.0 Flash (/vertex-ai/generative-ai/docs/gemini-v2).

Here is a simple text generation example.

Gen AI SDK for PythonGen AI SDK for Go... Gen AI SDK for Node.js... Gen AI SDK for Java...
(#gen-ai-sdk-for-python)

Install

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

To learn more, see the SDK reference documentation (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

client = genai.Client(http_options=HttpOptions(api_version="v1"))
response = client.models.generate_content(
    model="gemini-2.5-flash-preview-05-20",
    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:
# ...
```

Streaming and non-streaming responses

You can choose whether the model generates *streaming* responses or *non-streaming* responses. For streaming responses, you receive each response as soon as its output token is generated. For non-streaming responses, you receive all responses after all of the output tokens are generated.

Here is a streaming text generation example.

Python (#python)

Before trying this sample, follow the Python setup instructions in the Vertex AI quickstart using client libraries (/vertex-ai/docs/start/client-libraries). For more information, see the Vertex AI Python API reference documentation (/python/docs/reference/aiplatform/latest).

To authenticate to Vertex AI, set up Application Default Credentials. For more information, see Set up authentication for a local development environment (/docs/authentication/set-up-adc-local-dev-environment).

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

client = genai.Client(http_options=HttpOptions(api_version="v1"))
chat_session = client.chats.create(model="gemini-2.5-flash-preview-05-20")
```

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

What's next

- Learn how to send multimodal prompt requests:
  - Image understanding (/vertex-ai/generative-ai/docs/multimodal/image-understanding)
  - Video understanding (/vertex-ai/generative-ai/docs/multimodal/video-understanding)
  - Audio understanding (/vertex-ai/generative-ai/docs/multimodal/audio-understanding)
  - Document understanding (/vertex-ai/generative-ai/docs/multimodal/document-understanding)
- Learn about responsible AI best practices and Vertex AI's safety filters (/vertex-ai/generative-ai/docs/learn/responsible-ai).

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 2025-06-06 UTC.