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).

Grounding

Release Notes

In generative AI, grounding is the ability to connect model output to verifiable sources of information. If you provide models with access to specific data sources, then grounding tethers their output to these data and reduces the chances of inventing content.

With Vertex AI, you can ground model outputs in the following ways:

- Ground with Google Search - ground a model with publicly available web data.

- Ground to your own data - ground a model with your own data from Vertex AI Search as a data store (Preview (/products#product-launch-stages)).

For more information about grounding, see Grounding overview (/vertex-ai/generative-ai/docs/grounding/overview).

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) ▣

- 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)

Parameter list

See examples (#examples) for implementation details.

`GoogleSearchRetrieval`

Ground the response with public data.

Parameters

`google_search_retrieval`	Required: `Object`
	Ground with publicly available web data.

`Retrieval`

Ground the response with private data from Vertex AI Search as a data store. Defines a retrieval tool that the model can call to access external knowledge.

Parameters

`source`	Required: `VertexAISearch`
	Ground with Vertex AI Search data sources.

`VertexAISearch`

Parameters

`datastore`	Required: `string`
	Fully-qualified data store resource ID from Vertex AI Search, in the following format: `projects/{project}/locations/{location}/collections/default_collection/dataStores/{datastore}`

Examples

Ground response on public web data using Google Search

Ground the response with Google Search public data. Include the `google_search_retrieval` tool in the request. No additional parameters are required.

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
```

```python
from google import genai
from google.genai.types import (
    GenerateContentConfig,
    GoogleSearch,
    HttpOptions,
    Tool,
)

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

response = client.models.generate_content(
    model="gemini-2.5-flash-preview-05-20",
    contents="When is the next total solar eclipse in the United States?",
    config=GenerateContentConfig(
        tools=[
            # Use Google Search Tool
            Tool(google_search=GoogleSearch())
        ],
    ),
)

print(response.text)
```

```
# Example response:
# 'The next total solar eclipse in the United States will occur on ...'
```

Ground response on private data using Vertex AI Search

Preview

This product or 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 products and features are available "as is" and might have limited support. For more information, see the launch stage descriptions (/products#product-launch-stages).

Ground the response with data from a Vertex AI Search data store. For more information, see AI Applications (/vertex-ai-search-and-conversation).

Before you ground a response with private data, create a data store (/generative-ai-app-builder/docs/create-data-store-es) and a search app (/generative-ai-app-builder/docs/create-engine-es).

WARNING: For the time being, this "grounding" interface does not support Vertex AI Search "chunk mode".

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

```python
from google import genai
from google.genai.types import (
    GenerateContentConfig,
    HttpOptions,
    Retrieval,
    Tool,
    VertexAISearch,
)

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

# Load Data Store ID from Vertex AI Search
# datastore = "projects/111111111111/locations/global/collections/default_colle

response = client.models.generate_content(
    model="gemini-2.0-flash-001",
    contents="How do I make an appointment to renew my driver's license?",
```

```
    config=GenerateContentConfig(
        tools=[
            # Use Vertex AI Search Tool
            Tool(
                retrieval=Retrieval(
                    vertex_ai_search=VertexAISearch(
                        datastore=datastore,
                    )
                )
            )
        ],
    ),
)

print(response.text)
# Example response:
# 'The process for making an appointment to renew your driver's license varies
```

What's next

For detailed documentation, see the following:

- Grounding (/vertex-ai/generative-ai/docs/grounding/overview)

- Gemini API (/vertex-ai/generative-ai/docs/model-reference/gemini)