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).

Prompt management

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).

Vertex AI offers tooling to help manage prompt templates and prompt data. Prompt templates can be versioned and used in tandem with generative models on Vertex AI. Each prompt can be assembled and versioned in Vertex AI Studio or the Vertex AI SDK.

Vertex AI SDK includes the `vertexai.preview.prompts` module so that prompts can work with generative models. The `vertexai.preview.prompts` module supports the ability to define, save, and manage prompts for generating text with Gemini.

`Prompt`

The Prompt class (/vertex-ai/generative-ai/docs/reference/python/latest/vertexai.preview.prompts#prompt) represents a prompt that can be used to generate text with a Gemini method, which encapsulates the prompt data, variables, generation configuration, and other relevant information.

To create a `Prompt` object, use the `vertexai.preview.prompts.Prompt()` constructor. You can define the prompt data, variables, and other configurations within this object.

Create a local prompt and generate content

Vertex AI SDK for Python (#vertex-ai-sdk-for-python)

Vertex AI SDK for Python

```
import vertexai
from vertexai.preview import prompts
from vertexai.preview.prompts import Prompt

# from vertexai.generative_models import GenerationConfig, SafetySetting # Opti

# Initialize vertexai
vertexai.init(project=PROJECT_ID, location="us-central1")

# Create local Prompt
local_prompt = Prompt(
    prompt_name="movie-critic",
    prompt_data="Compare the movies {movie1} and {movie2}.",
    variables=[
        {"movie1": "The Lion King", "movie2": "Frozen"},
        {"movie1": "Inception", "movie2": "Interstellar"},
    ],
    model_name="gemini-2.0-flash-001",
    system_instruction="You are a movie critic. Answer in a short sentence.",
    # generation_config=GenerationConfig, # Optional,
    # safety_settings=SafetySetting, # Optional,
)

# Generate content using the assembled prompt for each variable set.
for i in range(len(local_prompt.variables)):
    response = local_prompt.generate_content(
        contents=local_prompt.assemble_contents(**local_prompt.variables[i])
    )
    print(response)

# Save a version
prompt1 = prompts.create_version(prompt=local_prompt)

print(prompt1)

# Example response
# Assembled prompt replacing: 1 instances of variable movie1, 1 instances of va
# Assembled prompt replacing: 1 instances of variable movie1, 1 instances of va
# Created prompt resource with id 12345678910.....
```

- **`project`**: Your project ID (/resource-manager/docs/creating-managing-projects#identifiers). You can find these IDs in the Google Cloud console welcome (https://console.cloud.google.com/welcome) page.

- `location`: See Vertex AI locations (/vertex-ai/docs/general/locations).
- `prompt_name`: The display name of the prompt created by the user, if stored in an online resource.
- `prompt_data`: A `PartsType` prompt, which can be a template with variables or a prompt with no variables.
- `variables`: A list of dictionaries containing the variable names and values.
- `generation_config`: A `GenerationConfig` object containing parameters for generation.
- `model_name`: Model Garden model resource name. Alternatively, a tuned model endpoint resource name can be provided. If no model is provided, the default latest model is used.
- `safety_settings`: A `SafetySetting` object containing safety settings for generation.
- `system_instruction`: A `PartsType` object representing the system instruction.

After the creation of a `Prompt` object, the prompt data and properties representing various configurations can be used to generate content.

Prompts also support function calling. See Introduction to function calling (/vertex-ai/generative-ai/docs/multimodal/function-calling) to learn more.

Save a prompt

To save a prompt to an online resource, which can be accessed in the Google Cloud console, use the `vertexai.preview.prompts.create_version()` method. This method takes a `Prompt` object as input and creates a new version of the prompt in the online store. A new `Prompt` object is returned which is associated with the online resource. Any updates made to a `Prompt` object are local until `create_version()` is called. The following code sample shows how to save a prompt:

Vertex AI SDK for Python (#vertex-ai-sdk-for-python)

```
from vertexai.preview import prompts

# Save Prompt to online resource.
# Returns a new Prompt object associated with the online resource.
prompt1 = prompts.create_version(prompt=prompt)
```

Load a saved prompt

To load a prompt that has been saved to the online resource, use the `vertexai.preview.prompts.get()` method. This method takes the prompt ID as input and returns the corresponding `Prompt` object. This code sample shows how to load a saved prompt:

Vertex AI SDK for Python (#vertex-ai-sdk-for-python)

```
from vertexai.preview import prompts

# Get prompt
prompt = prompts.get(prompt_id="123456789")
```

Retrieve prompt created in the Google Cloud console

To update a saved prompt, first load the prompt using the `get()` method (/vertex-ai/generative-ai/docs/reference/python/latest/vertexai.preview.prompts#vertexai_preview_prompts_get), modify its properties as needed, and then save the updated prompt using the `create_version()` method. This creates a new version of the prompt with the updated information.

Vertex AI SDK for Python (#vertex-ai-sdk-for-python)

```
from vertexai.preview import prompts
from vertexai.preview.prompts import Prompt

# Get prompt
prompt = prompts.get(prompt_id="123456789")

# Generate content using the assembled prompt (a prompt without variables)
prompt.generate_content(
  contents=prompt.assemble_contents()
)

# Update prompt (changes are local until create_version is called)
prompt.prompt_data = "new prompt"

# Save Prompt to online resource. Since the prompt is associated with a prompt
prompt1 = prompts.create_version(prompt=prompt)
```

List prompts

To see the display names and prompt IDs of all prompts saved in the current Google Cloud project, use the `list_prompts()` method (/vertex-ai/generative-ai/docs/reference/python/latest/vertexai.preview.prompts#vertexai_preview_prompts_list).

Vertex AI SDK for Python (#vertex-ai-sdk-for-python)

```
from vertexai.preview import prompts

prompts_metadata = prompts.list()

# Get a prompt from the list
prompt1 = prompts.get(prompt_id=prompts_metadata[0].prompt_id)
```

List prompt versions

To see the display names and version IDs of all prompt versions saved within the prompt, use the `list_versions()` method (/vertex-ai/generative-ai/docs/reference/python/latest/vertexai.preview.prompts#vertexai_preview_prompts_list_versions).

Vertex AI SDK for Python (#vertex-ai-sdk-for-python)

```
from vertexai.preview import prompts

prompt_versions_metadata = prompts.list_versions(prompt_id="123456789")

# Get a specific prompt version from the versions metadata list
prompt1 = prompts.get(
    prompt_id=prompt_versions_metadata[3].prompt_id,
    version_id=prompt_versions_metadata[3].version_id
)
```

Restore a prompt version

A prompt resource also contains version history that stores previous saved versions of the prompt. You can use the `restore_version()` method (/vertex-ai/generative-ai/docs/reference/python/latest/vertexai.preview.prompts#vertexai_preview_prompts_restore_version) to restore an older version as the latest version of the prompt. This returns PromptVersionMetadata that can be used with a `get()` call to fetch the newly restored version.

Vertex AI SDK for Python (#vertex-ai-sdk-for-python)

```
from vertexai.preview import prompts

# Restore to prompt version id 1 (original)
prompt_version_metadata = prompts.restore_version(prompt_id="123456789", versio

# Fetch the newly restored latest version of the prompt
prompt1 = prompts.get(prompt_id=prompt_version_metadata.prompt_id)
```

Delete a prompt

To delete the online resource associated with a prompt ID, use the `delete()` method (/vertex-ai/generative-ai/docs/reference/python/latest/vertexai.preview.prompts#vertexai_preview_prompts_delete) .

Vertex AI SDK for Python (#vertex-ai-sdk-for-python)

```
from vertexai.preview import prompts

prompts.delete(prompt_id="123456789")
```


Last updated 2025-06-06 UTC.