

Google Gen AI SDK

pypi v1.9.0

<https://github.com/googleapis/python-genai>

google-genai is an initial Python client library for interacting with Google's Generative AI APIs.

Google Gen AI Python SDK provides an interface for developers to integrate Google's generative models into their Python applications. It supports the `Gemini Developer API` and `Vertex AI APIs`.

Installation

```
pip install google-genai
```

Imports

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

Create a client

Please run one of the following code blocks to create a client for different services (Gemini Developer API or Vertex AI). Feel free to switch the client and run all the examples to see how it behaves under different APIs.

```
# Only run this block for Gemini Developer API
client = genai.Client(api_key='GEMINI_API_KEY')
```

```
# Only run this block for Vertex AI API
client = genai.Client(
    vertexai=True, project='your-project-id', location='us-central1'
)
```

(Optional) Using environment variables:

You can create a client by configuring the necessary environment variables. Configuration setup instructions depends on whether you're using the Gemini Developer API or the Gemini API in Vertex AI.

Gemini Developer API: Set `GOOGLE_API_KEY` as shown below:

```
export GOOGLE_API_KEY='your-api-key'
```

Skip to content

tex AI: Set `GOOGLE_GENAI_USE_VERTEXAI`, `GOOGLE_CLOUD_PROJECT` and `GOOGLE_CLOUD_LOCATION`, as shown below:

```
export GOOGLE_GENAI_USE_VERTEXAII=true
export GOOGLE_CLOUD_PROJECT='your-project-id'
export GOOGLE_CLOUD_LOCATION='us-central1'
```

```
client = genai.Client()
```

API Selection

By default, the SDK uses the beta API endpoints provided by Google to support preview features in the APIs. The stable API endpoints can be selected by setting the API version to `v1`.

To set the API version use `http_options`. For example, to set the API version to `v1` for Vertex AI:

```
client = genai.Client(
    vertexai=True,
    project='your-project-id',
    location='us-central1',
    http_options=types.HttpOptions(api_version='v1')
)
```

To set the API version to `v1alpha` for the Gemini Developer API:

```
# Only run this block for Gemini Developer API
client = genai.Client(
    api_key='GEMINI_API_KEY',
    http_options=types.HttpOptions(api_version='v1alpha')
)
```

Types

Parameter types can be specified as either dictionaries(`TypedDict`) or Pydantic Models. Pydantic model types are available in the `types` module.

Models

The `client.models` module exposes model inferencing and model getters.

Generate Content

with text content

```
response = client.models.generate_content(
    model='gemini-2.0-flash-001', contents='Why is the sky blue?'
)
    .text)
```

Skip to content

with uploaded file (Gemini Developer API only)

download the file in console.

```
!wget -q https://storage.googleapis.com/generativeai-downloads/data/a11.txt
```

python code.

```
file = client.files.upload(file='a11.txt')
response = client.models.generate_content(
    model='gemini-2.0-flash-001',
    contents=['Could you summarize this file?', file]
)
print(response.text)
```

How to structure *contents* argument for *generate_content*

The SDK always converts the inputs to the *contents* argument into *list[types.Content]*. The following shows some common ways to provide your inputs.

Provide a *list[types.Content]*

This is the canonical way to provide contents, SDK will not do any conversion.

Provide a *types.Content* instance

```
contents = types.Content(
    role='user',
    parts=[types.Part.from_text(text='Why is the sky blue?')]
)
```

SDK converts this to

```
[  
types.Content(  
    role='user',  
    parts=[types.Part.from_text(text='Why is the sky blue?')]  
)  
]
```

Provide a string

```
contents='Why is the sky blue?'
```

The SDK will assume this is a text part, and it converts this into the following:

Skip to content

```
[  
types.UserContent(  
    parts=[  
        types.Part.from_text(text='Why is the sky blue?')  
    ]  
)  
]
```

Where a `types.UserContent` is a subclass of `types.Content`, it sets the `role` field to be `user`.

Provide a list of string

The SDK assumes these are 2 text parts, it converts this into a single content, like the following:

```
[  
types.UserContent(  
    parts=[  
        types.Part.from_text(text='Why is the sky blue?'),  
        types.Part.from_text(text='Why is the cloud white?'),  
    ]  
)  
]
```

Where a `types.UserContent` is a subclass of `types.Content`, the `role` field in `types.UserContent` is fixed to be `user`.

Provide a function call part

```
contents = types.Part.from_function_call(  
    name='get_weather_by_location',  
    args={'location': 'Boston'})
```

The SDK converts a function call part to a content with a `model` role:

```
[  
types.ModelContent(  
    parts=[  
        types.Part.from_function_call(  
            name='get_weather_by_location',  
            args={'location': 'Boston'})  
    ]  
)  
]
```

Where a `types.ModelContent` is a subclass of `types.Content`, the `role` field in `types.ModelContent` is fixed to be `model`.

Skip to content

Provide a list of function call parts

```
contents = [
    types.Part.from_function_call(
        name='get_weather_by_location',
        args={'location': 'Boston'}
    ),
    types.Part.from_function_call(
        name='get_weather_by_location',
        args={'location': 'New York'}
    ),
]
```

The SDK converts a list of function call parts to the a content with a *model* role:

```
[
    types.ModelContent(
        parts=[
            types.Part.from_function_call(
                name='get_weather_by_location',
                args={'location': 'Boston'}
            ),
            types.Part.from_function_call(
                name='get_weather_by_location',
                args={'location': 'New York'}
            )
        ]
    )
]
```

Where a *types.ModelContent* is a subclass of *types.Content*, the *role* field in *types.ModelContent* is fixed to be *model*.

Provide a non function call part

```
contents = types.Part.from_uri(
    file_uri: 'gs://generativeai-downloads/images/scones.jpg',
    mime_type: 'image/jpeg',
)
```

The SDK converts all non function call parts into a content with a *user* role.

```
[
    types.UserContent(parts=[
        types.Part.from_uri(
            file_uri: 'gs://generativeai-downloads/images/scones.jpg',
            mime_type: 'image/jpeg',
        )
    ])
]
```

Skip to content

Provide a list of non function call parts

```
contents = [
    types.Part.from_text('What is this image about?'),
    types.Part.from_uri(
        file_uri: 'gs://generativeai-downloads/images/scones.jpg',
        mime_type: 'image/jpeg',
    )
]
```

The SDK will convert the list of parts into a content with a *user* role

```
[
    types.UserContent(
        parts=[
            types.Part.from_text('What is this image about?'),
            types.Part.from_uri(
                file_uri: 'gs://generativeai-downloads/images/scones.jpg',
                mime_type: 'image/jpeg',
            )
        ]
)
```

Mix types in contents

You can also provide a list of *types.ContentUnion*. The SDK leaves items of *types.Content* as is, it groups consecutive non function call parts into a single *types.UserContent*, and it groups consecutive function call parts into a single *types.ModelContent*.

If you put a list within a list, the inner list can only contain *types.PartUnion* items. The SDK will convert the inner list into a single *types.UserContent*.

System Instructions and Other Configs

The output of the model can be influenced by several optional settings available in `generate_content`'s config parameter. For example, the variability and length of the output can be influenced by the `temperature` and `max_output_tokens` respectively.

```
response = client.models.generate_content(
    model='gemini-2.0-flash-001',
    contents='high',
    config=types.GenerateContentConfig(
        system_instruction='I say high, you say low',
        max_output_tokens=3,
        temperature=0.3,
    ),
)
    .text)
```

Skip to content [_____](#)

Typed Config

All API methods support Pydantic types for parameters as well as dictionaries. You can get the type from `google.genai.types`.

```
response = client.models.generate_content(
    model='gemini-2.0-flash-001',
    contents=types.Part.from_text(text='Why is the sky blue?'),
    config=types.GenerateContentConfig(
        temperature=0,
        top_p=0.95,
        top_k=20,
        candidate_count=1,
        seed=5,
        max_output_tokens=100,
        stop_sequences=['STOP!'],
        presence_penalty=0.0,
        frequency_penalty=0.0,
    ),
)
print(response.text)
```

List Base Models

To retrieve tuned models, see: List Tuned Models

```
for model in client.models.list():
    print(model)
```

```
pager = client.models.list(config={'page_size': 10})
print(pager.page_size)
print(pager[0])
pager.next_page()
print(pager[0])
```

```
async for job in await client.aio.models.list():
    print(job)
```

```
async_pager = await client.aio.models.list(config={'page_size': 10})
print(async_pager.page_size)
print(async_pager[0])
await async_pager.next_page()
print(async_pager[0])
```

[Skip to content](#)

Safety Settings

```
response = client.models.generate_content(
    model='gemini-2.0-flash-001',
    contents='Say something bad.',
    config=types.GenerateContentConfig(
        safety_settings=[
            types.SafetySetting(
                category='HARM_CATEGORY_HATE_SPEECH',
                threshold='BLOCK_ONLY_HIGH',
            )
        ],
    ),
)
print(response.text)
```

Function Calling

You can pass a Python function directly and it will be automatically called and responded.

```
def get_current_weather(location: str) -> str:
    """Returns the current weather.

    Args:
        location: The city and state, e.g. San Francisco, CA
    """
    return 'sunny'

response = client.models.generate_content(
    model='gemini-2.0-flash-001',
    contents='What is the weather like in Boston?',
    config=types.GenerateContentConfig(
        tools=[get_current_weather],
    ),
)
print(response.text)
```

If you pass in a python function as a tool directly, and do not want automatic function calling, you can disable automatic function calling as follows:

With automatic function calling disabled, you will get a list of function call parts in the response:

If you don't want to use the automatic function support, you can manually declare the function and invoke it.

The following example shows how to declare a function and pass it as a tool. Then you will receive a `function_call` in the response.

[Skip to content](#)

```
function = types.FunctionDeclaration(  
    name='get_current_weather',  
    description='Get the current weather in a given location',  
    parameters=types.Schema(  
        type='OBJECT',  
        properties={  
            'location': types.Schema(  
                type='STRING',  
                description='The city and state, e.g. San Francisco, CA',  
            ),  
        },  
        required=['location'],  
    ),  
)  
  
tool = types.Tool(function_declarations=[function])  
  
response = client.models.generate_content(  
    model='gemini-2.0-flash-001',  
    contents='What is the weather like in Boston?',  
    config=types.GenerateContentConfig(  
        tools=[tool],  
    ),  
)  
print(response.function_calls[0])
```

After you receive the function call part from the model, you can invoke the function and get the function response. And then you can pass the function response to the model. The following example shows how to do it for a simple function invocation.

[Skip to content](#)

```
user_prompt_content = types.Content(
    role='user',
    parts=[types.Part.from_text(text='What is the weather like in Boston?')],
)
function_call_part = response.function_calls[0]
function_call_content = response.candidates[0].content

try:
    function_result = get_current_weather(
        **function_call_part.function_call.args
    )
    function_response = {'result': function_result}
except (
    Exception
) as e: # instead of raising the exception, you can let the model handle it
    function_response = {'error': str(e)}

function_response_part = types.Part.from_function_response(
    name=function_call_part.name,
    response=function_response,
)
function_response_content = types.Content(
    role='tool', parts=[function_response_part]
)

response = client.models.generate_content(
    model='gemini-2.0-flash-001',
    contents=[
        user_prompt_content,
        function_call_content,
        function_response_content,
    ],
    config=types.GenerateContentConfig(
        tools=[tool],
    ),
)
print(response.text)
```

If you configure function calling mode to be ANY, then the model will always return function call parts. If you also pass a python function as a tool, by default the SDK will perform automatic function calling until the remote calls exceed the maximum remote call for automatic function calling (default to 10 times).

If you'd like to disable automatic function calling in ANY mode:

Skip to content

```
def get_current_weather(location: str) -> str:
    """Returns the current weather.

Args:
    location: The city and state, e.g. San Francisco, CA
"""

    return "sunny"

response = client.models.generate_content(
    model="gemini-2.0-flash-001",
    contents="What is the weather like in Boston?",
    config=types.GenerateContentConfig(
        tools=[get_current_weather],
        automatic_function_calling=types.AutomaticFunctionCallingConfig(
            disable=True
        ),
        tool_config=types.ToolConfig(
            function_calling_config=types.FunctionCallingConfig(mode='ANY')
        ),
    ),
)
```

If you'd like to set x number of automatic function call turns, you can configure the maximum remote calls to be $x + 1$. Assuming you prefer 1 turn for automatic function calling:

```
def get_current_weather(location: str) -> str:
    """Returns the current weather.

Args:
    location: The city and state, e.g. San Francisco, CA
"""

    return "sunny"

response = client.models.generate_content(
    model="gemini-2.0-flash-001",
    contents="What is the weather like in Boston?",
    config=types.GenerateContentConfig(
        tools=[get_current_weather],
        automatic_function_calling=types.AutomaticFunctionCallingConfig(
            maximum_remote_calls=2
        ),
        tool_config=types.ToolConfig(
            function_calling_config=types.FunctionCallingConfig(mode='ANY')
        ),
    ),
)
```

JSON Response Schema

Skip to content [Provided as Pydantic Models.](#)

```
from pydantic import BaseModel

class CountryInfo(BaseModel):
    name: str
    population: int
    capital: str
    continent: str
    gdp: int
    official_language: str
    total_area_sq_mi: int

response = client.models.generate_content(
    model='gemini-2.0-flash-001',
    contents='Give me information for the United States.',
    config=types.GenerateContentConfig(
        response_mime_type='application/json',
        response_schema=CountryInfo,
    ),
)
print(response.text)
```

```
response = client.models.generate_content(
    model='gemini-2.0-flash-001',
    contents='Give me information for the United States.',
    config=types.GenerateContentConfig(
        response_mime_type='application/json',
        response_schema={
            'required': [
                'name',
                'population',
                'capital',
                'continent',
                'gdp',
                'official_language',
                'total_area_sq_mi',
            ],
            'properties': {
                'name': {'type': 'STRING'},
                'population': {'type': 'INTEGER'},
                'capital': {'type': 'STRING'},
                'continent': {'type': 'STRING'},
                'gdp': {'type': 'INTEGER'},
                'official_language': {'type': 'STRING'},
                'total_area_sq_mi': {'type': 'INTEGER'},
            },
            'type': 'OBJECT',
        },
    ),
)
```

Skip to content [.text](#))

Enum Response Schema

You can set `response_mime_type` to 'text/x.enum' to return one of those enum values as the response.

```
from enum import Enum

class InstrumentEnum(Enum):
    PERCUSSION = 'Percussion'
    STRING = 'String'
    WOODWIND = 'Woodwind'
    BRASS = 'Brass'
    KEYBOARD = 'Keyboard'

response = client.models.generate_content(
    model='gemini-2.0-flash-001',
    contents='What instrument plays multiple notes at once?',
    config={
        'response_mime_type': 'text/x.enum',
        'response_schema': InstrumentEnum,
    },
)
print(response.text)
```

You can also set `response_mime_type` to 'application/json', the response will be identical but in quotes.

```
class InstrumentEnum(Enum):
    PERCUSSION = 'Percussion'
    STRING = 'String'
    WOODWIND = 'Woodwind'
    BRASS = 'Brass'
    KEYBOARD = 'Keyboard'

response = client.models.generate_content(
    model='gemini-2.0-flash-001',
    contents='What instrument plays multiple notes at once?',
    config={
        'response_mime_type': 'application/json',
        'response_schema': InstrumentEnum,
    },
)
print(response.text)
```

Streaming

```
for chunk in client.models.generate_content_stream(
    model='gemini-2.0-flash-001', contents='Tell me a story in 300 words.'
):
    print(chunk['text', end=' ')]
```

Skip to content [\[\]](#)

If your image is stored in Google Cloud Storage, you can use the `from_uri` class method to create a `Part` object.

```
for chunk in client.models.generate_content_stream(
    model='gemini-2.0-flash-001',
    contents=[
        'What is this image about?',
        types.Part.from_uri(
            file_uri='gs://generativeai-downloads/images/scones.jpg',
            mime_type='image/jpeg',
        ),
    ],
):
    print(chunk.text, end='')
```

If your image is stored in your local file system, you can read it in as bytes data and use the `from_bytes` class method to create a `Part` object.

```
YOUR_IMAGE_PATH = 'your_image_path'
YOUR_IMAGE_MIME_TYPE = 'your_image_mime_type'
with open(YOUR_IMAGE_PATH, 'rb') as f:
    image_bytes = f.read()

for chunk in client.models.generate_content_stream(
    model='gemini-2.0-flash-001',
    contents=[
        'What is this image about?',
        types.Part.from_bytes(data=image_bytes, mime_type=YOUR_IMAGE_MIME_TYPE),
    ],
):
    print(chunk.text, end='')
```

Async

`client.aio` exposes all the analogous `async` methods that are available on `client`

For example, `client.aio.models.generate_content` is the `async` version of `client.models.generate_content`

```
response = await client.aio.models.generate_content(
    model='gemini-2.0-flash-001', contents='Tell me a story in 300 words.'
)

print(response.text)
```

Streaming

```
async for chunk in await client.aio.models.generate_content_stream(
    model='gemini-2.0-flash-001', contents='Tell me a story in 300 words.'
```

Skip to content `<.text, end=''`

Count Tokens and Compute Tokens

```
response = client.models.count_tokens(
    model='gemini-2.0-flash-001',
    contents='why is the sky blue?',
)
print(response)
```

Compute tokens is only supported in Vertex AI.

```
response = client.models.compute_tokens(
    model='gemini-2.0-flash-001',
    contents='why is the sky blue?',
)
print(response)
```

Async

```
response = await client.aio.models.count_tokens(
    model='gemini-2.0-flash-001',
    contents='why is the sky blue?',
)
print(response)
```

Embed Content

```
response = client.models.embed_content(
    model='text-embedding-004',
    contents='why is the sky blue?',
)
print(response)
```

```
# multiple contents with config
response = client.models.embed_content(
    model='text-embedding-004',
    contents=['why is the sky blue?', 'What is your age?'],
    config=types.EmbedContentConfig(output_dimensionality=10),
)
print(response)
```

Imagen

Support for generate image in Gemini Developer API is behind an allowlist

Skip to content

```
# Generate Image
response1 = client.models.generate_images(
    model='imagen-3.0-generate-002',
    prompt='An umbrella in the foreground, and a rainy night sky in the background',
    config=types.GenerateImagesConfig(
        number_of_images=1,
        include_rai_reason=True,
        output_mime_type='image/jpeg',
    ),
)
response1.generated_images[0].image.show()
```

Upscale image is only supported in Vertex AI.

```
# Upscale the generated image from above
response2 = client.models.upscale_image(
    model='imagen-3.0-generate-002',
    image=response1.generated_images[0].image,
    upscale_factor='x2',
    config=types.UpscaleImageConfig(
        include_rai_reason=True,
        output_mime_type='image/jpeg',
    ),
)
response2.generated_images[0].image.show()
```

Edit image uses a separate model from generate and upscale.

Edit image is only supported in Vertex AI.

[Skip to content](#)

```
# Edit the generated image from above
from google.genai.types import RawReferenceImage, MaskReferenceImage

raw_ref_image = RawReferenceImage(
    reference_id=1,
    reference_image=response1.generated_images[0].image,
)

# Model computes a mask of the background
mask_ref_image = MaskReferenceImage(
    reference_id=2,
    config=types.MaskReferenceConfig(
        mask_mode='MASK_MODE_BACKGROUND',
        mask_dilation=0,
    ),
)

response3 = client.models.edit_image(
    model='imagen-3.0-capability-001',
    prompt='Sunlight and clear sky',
    reference_images=[raw_ref_image, mask_ref_image],
    config=types>EditImageConfig(
        edit_mode='EDIT_MODE_INPAINT_INSERTION',
        number_of_images=1,
        include_rai_reason=True,
        output_mime_type='image/jpeg',
    ),
)
response3.generated_images[0].image.show()
```

Veo

Support for generate videos in Vertex and Gemini Developer API is behind an allowlist

```
# Create operation
operation = client.models.generate_videos(
    model='veo-2.0-generate-001',
    prompt='A neon hologram of a cat driving at top speed',
    config=types.GenerateVideosConfig(
        number_of_videos=1,
        fps=24,
        duration_seconds=5,
        enhance_prompt=True,
    ),
)

# Poll operation
while not operation.done:
    time.sleep(20)
    operation = client.operations.get(operation)
```

Skip to content

```
    ion.result.generated_videos[0].video
    video.show()
```

Chats

Create a chat session to start a multi-turn conversations with the model.

Send Message

```
chat = client.chats.create(model='gemini-2.0-flash-001')
response = chat.send_message('tell me a story')
print(response.text)
```

Streaming

```
chat = client.chats.create(model='gemini-2.0-flash-001')
for chunk in chat.send_message_stream('tell me a story'):
    print(chunk.text, end='')
```

Async

```
chat = client.aio.chats.create(model='gemini-2.0-flash-001')
response = await chat.send_message('tell me a story')
print(response.text)
```

Async Streaming

```
chat = client.aio.chats.create(model='gemini-2.0-flash-001')
async for chunk in await chat.send_message_stream('tell me a story'):
    print(chunk.text, end='')
```

Files

Files are only supported in Gemini Developer API.

```
gsutil cp gs://cloud-samples-data/generative-ai/pdf/2312.11805v3.pdf .
gsutil cp gs://cloud-samples-data/generative-ai/pdf/2403.05530.pdf .
```

Upload

```
file1 = client.files.upload(file='2312.11805v3.pdf')
file2 = client.files.upload(file='2403.05530.pdf')
```

Skip to content

```
print(file1)
```

Get

```
file1 = client.files.upload(file='2312.11805v3.pdf')
file_info = client.files.get(name=file1.name)
```

Delete

```
file3 = client.files.upload(file='2312.11805v3.pdf')

client.files.delete(name=file3.name)
```

Caches

`client.caches` contains the control plane APIs for cached content

Create

```
if client.vertexai:
    file_uris = [
        'gs://cloud-samples-data/generative-ai/pdf/2312.11805v3.pdf',
        'gs://cloud-samples-data/generative-ai/pdf/2403.05530.pdf',
    ]
else:
    file_uris = [file1.uri, file2.uri]

cached_content = client.caches.create(
    model='gemini-1.5-pro-002',
    config=types.CreateCachedContentConfig(
        contents=[
            types.Content(
                role='user',
                parts=[
                    types.Part.from_uri(
                        file_uri=file_uris[0], mime_type='application/pdf'
                    ),
                    types.Part.from_uri(
                        file_uri=file_uris[1],
                        mime_type='application/pdf',
                    ),
                ],
            ),
        ],
        system_instruction='What is the sum of the two pdfs?',
        display_name='test cache',
        ttl='3600s',
    ),
)
```

[Skip to content](#)

Get

```
cached_content = client.caches.get(name=cached_content.name)
```

Generate Content

```
response = client.models.generate_content(
    model='gemini-1.5-pro-002',
    contents='Summarize the pdfs',
    config=types.GenerateContentConfig(
        cached_content=cached_content.name,
    ),
)
print(response.text)
```

Tunings

`client.tunings` contains tuning job APIs and supports supervised fine tuning through `tune`.

Tune

- Vertex AI supports tuning from GCS source
- Gemini Developer API supports tuning from inline examples

```
if client.vertexai:
    model = 'gemini-1.5-pro-002'
    training_dataset = types.TuningDataset(
        gcs_uri='gs://cloud-samples-data/ai-platform/generative_ai/gemini-1_5/text/sft_trai
    )
else:
    model = 'models/gemini-1.0-pro-001'
    training_dataset = types.TuningDataset(
        examples=[
            types.TuningExample(
                text_input=f'Input text {i}',
                output=f'Output text {i}',
            )
            for i in range(5)
        ],
    )
```

Skip to content

```
tuning_job = client.tunings.tune(
    base_model=model,
    training_dataset=training_dataset,
    config=types.CreateTuningJobConfig(
        epoch_count=1, tuned_model_display_name='test_dataset_examples model'
    ),
)
print(tuning_job)
```

Get Tuning Job

```
tuning_job = client.tunings.get(name=tuning_job.name)
print(tuning_job)
```

```
import time

running_states = set(
    [
        'JOB_STATE_PENDING',
        'JOB_STATE_RUNNING',
    ]
)

while tuning_job.state in running_states:
    print(tuning_job.state)
    tuning_job = client.tunings.get(name=tuning_job.name)
    time.sleep(10)
```

```
response = client.models.generate_content(
    model=tuning_job.tuned_model.endpoint,
    contents='why is the sky blue?',
)
print(response.text)
```

Get Tuned Model

```
tuned_model = client.models.get(model=tuning_job.tuned_model.model)
print(tuned_model)
```

List Tuned Models

To retrieve base models, see: [List Base Models](#)

Skip to content `client.models.list(config={'page_size': 10, 'query_base': False}): 1`

```
pager = client.models.list(config={'page_size': 10, 'query_base': False})
print(pager.page_size)
print(pager[0])
pager.next_page()
print(pager[0])
```

```
async for job in await client.aio.models.list(config={'page_size': 10, 'query_base': False}):
    print(job)
```

```
async_pager = await client.aio.models.list(config={'page_size': 10, 'query_base': False})
print(async_pager.page_size)
print(async_pager[0])
await async_pager.next_page()
print(async_pager[0])
```

Update Tuned Model

```
model = pager[0]

model = client.models.update(
    model=model.name,
    config=types.UpdateModelConfig(
        display_name='my tuned model', description='my tuned model description'
    ),
)

print(model)
```

List Tuning Jobs

```
for job in client.tunings.list(config={'page_size': 10}):
    print(job)
```

```
pager = client.tunings.list(config={'page_size': 10})
print(pager.page_size)
print(pager[0])
pager.next_page()
print(pager[0])
```

```
async for job in await client.aio.tunings.list(config={'page_size': 10}):
    print(job)
```

Skip to content

```
async_page = await client.aio.tunings.list(config={'page_size': 10})
print(async_page.page_size)
print(async_page[0])
await async_page.next_page()
print(async_page[0])
```

Batch Prediction

Only supported in Vertex AI.

Create

```
# Specify model and source file only, destination and job display name will be auto-populated
job = client.batches.create(
    model='gemini-1.5-flash-002',
    src='bq://my-project.my-dataset.my-table',
)
job
```

```
# Get a job by name
job = client.batches.get(name=job.name)

job.state
```

```
completed_states = set(
    [
        'JOB_STATE_SUCCEEDED',
        'JOB_STATE_FAILED',
        'JOB_STATE_CANCELLED',
        'JOB_STATE_PAUSED',
    ]
)

while job.state not in completed_states:
    print(job.state)
    job = client.batches.get(name=job.name)
    time.sleep(30)

job
```

List

[client.batches.list\(config=types.ListBatchJobsConfig\(page_size=10\)\):](#)
Skip to content

```

pager = client.batches.list(config=types.ListBatchJobsConfig(page_size=10))
print(pager.page_size)
print(pager[0])
pager.next_page()
print(pager[0])

```

```

async for job in await client.aio.batches.list(
    config=types.ListBatchJobsConfig(page_size=10)
):
    print(job)

```

```

async_pager = await client.aio.batches.list(
    config=types.ListBatchJobsConfig(page_size=10)
)
print(async_pager.page_size)
print(async_pager[0])
await async_pager.next_page()
print(async_pager[0])

```

Delete

```

# Delete the job resource
delete_job = client.batches.delete(name=job.name)

delete_job

```

Error Handling

To handle errors raised by the model, the SDK provides this [APIError] (<https://github.com/googleapis/python-genai/blob/main/google/genai/errors.py>) class.

```

try:
    client.models.generate_content(
        model="invalid-model-name",
        contents="What is your name?",
    )
except errors.APIError as e:
    print(e.code) # 404
    print(e.message)

```

[Skip to content](#)

Reference

- [Submodules](#)
- [genai.client module](#)
 - [AsyncClient](#)
 - [AsyncClient.batches](#)
 - [AsyncClient.caches](#)
 - [AsyncClient.chats](#)
 - [AsyncClient.files](#)
 - [AsyncClient.live](#)
 - [AsyncClient.models](#)
 - [AsyncClient.operations](#)
 - [AsyncClient.tunings](#)
 - [Client](#)
 - [Client.api_key](#)
 - [Client.vertexai](#)
 - [Client.credentials](#)
 - [Client.project](#)
 - [Client.location](#)
 - [Client.debug_config](#)
 - [Client.http_options](#)
 - [Client.aio](#)
 - [Client.batches](#)
 - [Client.caches](#)
 - [Client.chats](#)
 - [Client.files](#)
 - [Client.models](#)
 - [Client.operations](#)
 - [Client.tunings](#)
 - [Client.vertexai](#)
 - [DebugConfig](#)
 - [DebugConfig.client_mode](#)
 - [DebugConfig.replay_id](#)
 - [DebugConfig.replays_directory](#)

[Skip to content](#)

- [genai.batches module](#)

- [AsyncBatches](#)
 - [AsyncBatches.cancel\(\)](#)
 - [AsyncBatches.create\(\)](#)
 - [AsyncBatches.delete\(\)](#)
 - [AsyncBatches.get\(\)](#)
 - [AsyncBatches.list\(\)](#)

- [Batches](#)
 - [Batches.cancel\(\)](#)
 - [Batches.create\(\)](#)
 - [Batches.delete\(\)](#)
 - [Batches.get\(\)](#)
 - [Batches.list\(\)](#)

- [genai.caches module](#)

- [AsyncCaches](#)
 - [AsyncCaches.create\(\)](#)
 - [AsyncCaches.delete\(\)](#)
 - [AsyncCaches.get\(\)](#)
 - [AsyncCaches.list\(\)](#)
 - [AsyncCaches.update\(\)](#)

- [Caches](#)
 - [Caches.create\(\)](#)
 - [Caches.delete\(\)](#)
 - [Caches.get\(\)](#)
 - [Caches.list\(\)](#)
 - [Caches.update\(\)](#)

- [genai.chats module](#)

- [AsyncChat](#)
 - [AsyncChat.send_message\(\)](#)
 - [AsyncChat.send_message_stream\(\)](#)
- [AsyncChats](#)
 - [AsyncChats.create\(\)](#)
- [Chat](#)
 - [Chat.send_message\(\)](#)
 - [Chat.send_message_stream\(\)](#)
- [Chats](#)
 - [Chats.create\(\)](#)

[Skip to content](#)

- [genai.files module](#)

- [AsyncFiles](#)
 - [AsyncFiles.delete\(\)](#)
 - [AsyncFiles.download\(\)](#)
 - [AsyncFiles.get\(\)](#)
 - [AsyncFiles.list\(\)](#)
 - [AsyncFiles.upload\(\)](#)

- [Files](#)
 - [Files.delete\(\)](#)
 - [Files.download\(\)](#)
 - [Files.get\(\)](#)
 - [Files.list\(\)](#)
 - [Files.upload\(\)](#)

- [genai.live module](#)

- [AsyncLive](#)
 - [AsyncLive.connect\(\)](#)
- [AsyncSession](#)
 - [AsyncSession.close\(\)](#)
 - [AsyncSession.receive\(\)](#)
 - [AsyncSession.send\(\)](#)
 - [AsyncSession.start_stream\(\)](#)

[Skip to content](#)

- [genai.models module](#)

- [AsyncModels](#)
 - [AsyncModels.compute_tokens\(\)](#)
 - [AsyncModels.count_tokens\(\)](#)
 - [AsyncModels.delete\(\)](#)
 - [AsyncModels.edit_image\(\)](#)
 - [AsyncModels.embed_content\(\)](#)
 - [AsyncModels.generate_content\(\)](#)
 - [AsyncModels.generate_content_stream\(\)](#)
 - [AsyncModels.generate_images\(\)](#)
 - [AsyncModels.generate_videos\(\)](#)
 - [AsyncModels.get\(\)](#)
 - [AsyncModels.list\(\)](#)
 - [AsyncModels.update\(\)](#)
 - [AsyncModels.upscale_image\(\)](#)
- [Models](#)
 - [Models.compute_tokens\(\)](#)
 - [Models.count_tokens\(\)](#)
 - [Models.delete\(\)](#)
 - [Models.edit_image\(\)](#)
 - [Models.embed_content\(\)](#)
 - [Models.generate_content\(\)](#)
 - [Models.generate_content_stream\(\)](#)
 - [Models.generate_images\(\)](#)
 - [Models.generate_videos\(\)](#)
 - [Models.get\(\)](#)
 - [Models.list\(\)](#)
 - [Models.update\(\)](#)
 - [Models.upscale_image\(\)](#)

- [genai.tunings module](#)

- [AsyncTunings](#)
 - [AsyncTunings.get\(\)](#)
 - [AsyncTunings.list\(\)](#)
 - [AsyncTunings.tune\(\)](#)
- [Tunings](#)
 - [Tunings.get\(\)](#)
 - [Tunings.list\(\)](#)
 - [Tunings.tune\(\)](#)

[Skip to content](#)

- [genai.types module](#)

- [AdapterSize](#)
 - `AdapterSize.ADAPTER_SIZE_EIGHT`
 - `AdapterSize.ADAPTER_SIZE_FOUR`
 - `AdapterSize.ADAPTER_SIZE_ONE`
 - `AdapterSize.ADAPTER_SIZE_SIXTEEN`
 - `AdapterSize.ADAPTER_SIZE_THIRTY_TWO`
 - `AdapterSize.ADAPTER_SIZE_UNSPECIFIED`
- [AutomaticFunctionCallingConfig](#)
 - `AutomaticFunctionCallingConfig.disable`
 - `AutomaticFunctionCallingConfig.ignore_call_history`
 - `AutomaticFunctionCallingConfig.maximum_remote_calls`
- [AutomaticFunctionCallingConfigDict](#)
 - `AutomaticFunctionCallingConfigDict.disable`
 - `AutomaticFunctionCallingConfigDict.ignore_call_history`
 - `AutomaticFunctionCallingConfigDict.maximum_remote_calls`
- [BatchJob](#)
 - `BatchJob.create_time`
 - `BatchJob.dest`
 - `BatchJob.display_name`
 - `BatchJob.end_time`
 - `BatchJob.error`
 - `BatchJob.model`
 - `BatchJob.name`
 - `BatchJob.src`
 - `BatchJob.start_time`
 - `BatchJob.state`
 - `BatchJob.update_time`
- [BatchJobDestination](#)
 - `BatchJobDestination.biggquery_uri`
 - `BatchJobDestination.format`
 - `BatchJobDestination.gcs_uri`
- [BatchJobDestinationDict](#)
 - `BatchJobDestinationDict.biggquery_uri`
 - `BatchJobDestinationDict.format`
 - `BatchJobDestinationDict.gcs_uri`

[Skip to content](#)

- BatchJobDict
 - `BatchJobDict.create_time`
 - `BatchJobDict.dest`
 - `BatchJobDict.display_name`
 - `BatchJobDict.end_time`
 - `BatchJobDict.error`
 - `BatchJobDict.model`
 - `BatchJobDict.name`
 - `BatchJobDict.src`
 - `BatchJobDict.start_time`
 - `BatchJobDict.state`
 - `BatchJobDict.update_time`
- BatchJobSource
 - `BatchJobSource.bigquery_uri`
 - `BatchJobSource.format`
 - `BatchJobSource.gcs_uri`
- BatchJobSourceDict
 - `BatchJobSourceDict.bigquery_uri`
 - `BatchJobSourceDict.format`
 - `BatchJobSourceDict.gcs_uri`
- Blob
 - `Blob.data`
 - `Blob.mime_type`
- BlobDict
 - `BlobDict.data`
 - `BlobDict.mime_type`
- BlockedReason
 - `BlockedReason.BLOCKED_REASON_UNSPECIFIED`
 - `BlockedReason.BLOCKLIST`
 - `BlockedReason.OTHER`
 - `BlockedReason.PROHIBITED_CONTENT`
 - `BlockedReason.SAFETY`
- CachedContent
 - `CachedContent.create_time`
 - `CachedContent.display_name`
 - `CachedContent.expire_time`
 - `CachedContent.model`
 - `CachedContent.name`
 - `CachedContent.update_time`
 - `CachedContent.usage_metadata`

[Skip to content](#)

- CachedContentDict
 - `CachedContentDict.create_time`
 - `CachedContentDict.display_name`
 - `CachedContentDict.expire_time`
 - `CachedContentDict.model`
 - `CachedContentDict.name`
 - `CachedContentDict.update_time`
 - `CachedContentDict.usage_metadata`
- CachedContentUsageMetadata
 - `CachedContentUsageMetadata.audio_duration_seconds`
 - `CachedContentUsageMetadata.image_count`
 - `CachedContentUsageMetadata.text_count`
 - `CachedContentUsageMetadata.total_token_count`
 - `CachedContentUsageMetadata.video_duration_seconds`
- CachedContentUsageMetadataDict
 - `CachedContentUsageMetadataDict.audio_duration_seconds`
 - `CachedContentUsageMetadataDict.image_count`
 - `CachedContentUsageMetadataDict.text_count`
 - `CachedContentUsageMetadataDict.total_token_count`
 - `CachedContentUsageMetadataDict.video_duration_seconds`
- CancelBatchJobConfig
 - `CancelBatchJobConfig.http_options`
- CancelBatchJobConfigDict
 - `CancelBatchJobConfigDict.http_options`
- Candidate
 - `Candidate.avg_logprobs`
 - `Candidate.citation_metadata`
 - `Candidate.content`
 - `Candidate.finish_message`
 - `Candidate.finish_reason`
 - `Candidate.grounding_metadata`
 - `Candidate.index`
 - `Candidate.logprobs_result`
 - `Candidate.safety_ratings`
 - `Candidate.token_count`

[Skip to content](#)

- CandidateDict
 - CandidateDict.avg_logprobs
 - CandidateDict.citation_metadata
 - CandidateDict.content
 - CandidateDict.finish_message
 - CandidateDict.finish_reason
 - CandidateDict.grounding_metadata
 - CandidateDict.index
 - CandidateDict.logprobs_result
 - CandidateDict.safety_ratings
 - CandidateDict.token_count
- Citation
 - Citation.end_index
 - Citation.license
 - Citation.publication_date
 - Citation.start_index
 - Citation.title
 - Citation.uri
- CitationDict
 - CitationDict.end_index
 - CitationDict.license
 - CitationDict.publication_date
 - CitationDict.start_index
 - CitationDict.title
 - CitationDict.uri
- CitationMetadata
 - CitationMetadata.citations
- CitationMetadataDict
 - CitationMetadataDict.citations
- CodeExecutionResult
 - CodeExecutionResult.outcome
 - CodeExecutionResult.output
- CodeExecutionResultDict
 - CodeExecutionResultDict.outcome
 - CodeExecutionResultDict.output
- ComputeTokensConfig
 - ComputeTokensConfig.http_options
- ComputeTokensConfigDict
 - ComputeTokensConfigDict.http_options
- ComputeTokensResponse
 - ComputeTokensResponse.tokens_info
- ComputeTokensResponseDict
 - ComputeTokensResponseDict.tokens_info

Skip to content [tokensResponseDict.tokens_info](#)

- Content
 - Content.parts
 - Content.role
- ContentDict
 - ContentDict.parts
 - ContentDict.role
- ContentEmbedding
 - ContentEmbedding.statistics
 - ContentEmbedding.values
- ContentEmbeddingDict
 - ContentEmbeddingDict.statistics
- ContentEmbeddingStatistics
 - ContentEmbeddingStatistics.token_count
 - ContentEmbeddingStatistics.truncated
- ContentEmbeddingStatisticsDict
 - ContentEmbeddingStatisticsDict.token_count
 - ContentEmbeddingStatisticsDict.truncated
- ControlReferenceConfig
 - ControlReferenceConfig.control_type
 - ControlReferenceConfig.enable_control_image_computation
- ControlReferenceConfigDict
 - ControlReferenceConfigDict.control_type
 - ControlReferenceConfigDict.enable_control_image_computation
- ControlReferenceImage
 - ControlReferenceImage.config
 - ControlReferenceImage.control_image_config
 - ControlReferenceImage.reference_id
 - ControlReferenceImage.reference_image
 - ControlReferenceImage.reference_type
- ControlReferenceImageDict
 - ControlReferenceImageDict.config
 - ControlReferenceImageDict.reference_id
 - ControlReferenceImageDict.reference_image
 - ControlReferenceImageDict.reference_type
- ControlReferenceType
 - ControlReferenceType.CONTROL_TYPE_CANNY
 - ControlReferenceType.CONTROL_TYPE_DEFAULT
 - ControlReferenceType.CONTROL_TYPE_FACE_MESH
 - ControlReferenceType.CONTROL_TYPE_SCRIBBLE
- CountTokensConfig
 - CountTokensConfig.generation_config
 - CountTokensConfig.http_options

Skip to content `ensConfig.system_instruction`
 `ensConfig.tools`

- CountTokensConfigDict
 - CountTokensConfigDict.generation_config
 - CountTokensConfigDict.http_options
 - CountTokensConfigDict.system_instruction
 - CountTokensConfigDict.tools
- CountTokensResponse
 - CountTokensResponse.cached_content_token_count
 - CountTokensResponse.total_tokens
- CountTokensResponseDict
 - CountTokensResponseDict.cached_content_token_count
 - CountTokensResponseDict.total_tokens
- CreateBatchJobConfig
 - CreateBatchJobConfig.dest
 - CreateBatchJobConfig.display_name
 - CreateBatchJobConfig.http_options
- CreateBatchJobConfigDict
 - CreateBatchJobConfigDict.dest
 - CreateBatchJobConfigDict.display_name
 - CreateBatchJobConfigDict.http_options
- CreateCachedContentConfig
 - CreateCachedContentConfig.contents
 - CreateCachedContentConfig.display_name
 - CreateCachedContentConfig.expire_time
 - CreateCachedContentConfig.http_options
 - CreateCachedContentConfig.system_instruction
 - CreateCachedContentConfig.tool_config
 - CreateCachedContentConfig.tools
 - CreateCachedContentConfig.ttl
- CreateCachedContentConfigDict
 - CreateCachedContentConfigDict.contents
 - CreateCachedContentConfigDict.display_name
 - CreateCachedContentConfigDict.expire_time
 - CreateCachedContentConfigDict.http_options
 - CreateCachedContentConfigDict.system_instruction
 - CreateCachedContentConfigDict.tool_config
 - CreateCachedContentConfigDict.tools
 - CreateCachedContentConfigDict.ttl
- CreateFileConfig
 - CreateFileConfig.http_options
- CreateFileConfigDict
 - CreateFileConfigDict.http_options
- CreateFileResponse
 - CreateFileResponseDict.http_headers

Skip to content leResponse.http_headers
 .esponseDict
 ▪ CreateFileResponseDict.http_headers

- `CreateTuningJobConfig`
 - `CreateTuningJobConfig.adapter_size`
 - `CreateTuningJobConfig.batch_size`
 - `CreateTuningJobConfig.description`
 - `CreateTuningJobConfig.epoch_count`
 - `CreateTuningJobConfig.http_options`
 - `CreateTuningJobConfig.learning_rate`
 - `CreateTuningJobConfig.learning_rate_multiplier`
 - `CreateTuningJobConfig.tuned_model_display_name`
 - `CreateTuningJobConfig.validation_dataset`
- `CreateTuningJobConfigDict`
 - `CreateTuningJobConfigDict.adapter_size`
 - `CreateTuningJobConfigDict.batch_size`
 - `CreateTuningJobConfigDict.description`
 - `CreateTuningJobConfigDict.epoch_count`
 - `CreateTuningJobConfigDict.http_options`
 - `CreateTuningJobConfigDict.learning_rate`
 - `CreateTuningJobConfigDict.learning_rate_multiplier`
 - `CreateTuningJobConfigDict.tuned_model_display_name`
 - `CreateTuningJobConfigDict.validation_dataset`
- `DatasetDistribution`
 - `DatasetDistribution.buckets`
 - `DatasetDistribution.max`
 - `DatasetDistribution.mean`
 - `DatasetDistribution.median`
 - `DatasetDistribution.min`
 - `DatasetDistribution.p5`
 - `DatasetDistribution.p95`
 - `DatasetDistribution.sum`
- `DatasetDistributionDict`
 - `DatasetDistributionDict.buckets`
 - `DatasetDistributionDict.max`
 - `DatasetDistributionDict.mean`
 - `DatasetDistributionDict.median`
 - `DatasetDistributionDict.min`
 - `DatasetDistributionDict.p5`
 - `DatasetDistributionDict.p95`
 - `DatasetDistributionDict.sum`
- `DatasetDistributionDistributionBucket`
 - `DatasetDistributionDistributionBucket.count`
 - `DatasetDistributionDistributionBucket.left`
 - `DatasetDistributionDistributionBucket.right`

[Skip to content](#)

- `DatasetDistributionDistributionBucketDict`
 - `DatasetDistributionDistributionBucketDict.count`
 - `DatasetDistributionDistributionBucketDict.left`
 - `DatasetDistributionDistributionBucketDict.right`
- `DatasetStats`
 - `DatasetStats.total_billable_character_count`
 - `DatasetStats.total_tuning_character_count`
 - `DatasetStats.tuning_dataset_example_count`
 - `DatasetStats.tuning_step_count`
 - `DatasetStats.user_dataset_examples`
 - `DatasetStats.user_input_token_distribution`
 - `DatasetStats.user_message_per_example_distribution`
 - `DatasetStats.user_output_token_distribution`
- `DatasetStatsDict`
 - `DatasetStatsDict.total_billable_character_count`
 - `DatasetStatsDict.total_tuning_character_count`
 - `DatasetStatsDict.tuning_dataset_example_count`
 - `DatasetStatsDict.tuning_step_count`
 - `DatasetStatsDict.user_dataset_examples`
 - `DatasetStatsDict.user_input_token_distribution`
 - `DatasetStatsDict.user_message_per_example_distribution`
 - `DatasetStatsDict.user_output_token_distribution`
- `DeleteBatchJobConfig`
 - `DeleteBatchJobConfig.http_options`
- `DeleteBatchJobConfigDict`
 - `DeleteBatchJobConfigDict.http_options`
- `DeleteCachedContentConfig`
 - `DeleteCachedContentConfig.http_options`
- `DeleteCachedContentConfigDict`
 - `DeleteCachedContentConfigDict.http_options`
- `DeleteCachedContentResponse`
- `DeleteCachedContentResponseDict`
- `DeleteFileConfig`
 - `DeleteFileConfig.http_options`
- `DeleteFileConfigDict`
 - `DeleteFileConfigDict.http_options`
- `DeleteFileResponse`
- `DeleteFileResponseDict`
- `DeleteModelConfig`
 - `DeleteModelConfig.http_options`
- `DeleteModelConfigDict`
 - `DeleteModelConfigDict.http_options`

Skip to content Response
 .ResponseDict

- DeleteResourceJob
 - DeleteResourceJob.done
 - DeleteResourceJob.error
 - DeleteResourceJob.name
- DeleteResourceJobDict
 - DeleteResourceJobDict.done
 - DeleteResourceJobDict.error
 - DeleteResourceJobDict.name
- DeploymentResourcesType
 - DeploymentResourcesType.AUTOMATIC_RESOURCES
 - DeploymentResourcesType.DEDICATED_RESOURCES
 - DeploymentResourcesType.DEPLOYMENT_RESOURCES_TYPE_UNSPECIFIED
 - DeploymentResourcesType.SHARED_RESOURCES
- DistillationDataStats
 - DistillationDataStats.training_dataset_stats
- DistillationDataStatsDict
 - DistillationDataStatsDict.training_dataset_stats
- DistillationHyperParameters
 - DistillationHyperParameters.adapter_size
 - DistillationHyperParameters.epoch_count
 - DistillationHyperParameters.learning_rate_multiplier
- DistillationHyperParametersDict
 - DistillationHyperParametersDict.adapter_size
 - DistillationHyperParametersDict.epoch_count
 - DistillationHyperParametersDict.learning_rate_multiplier
- DistillationSpec
 - DistillationSpec.base_teacher_model
 - DistillationSpec.hyper_parameters
 - DistillationSpec.pipeline_root_directory
 - DistillationSpec.student_model
 - DistillationSpec.training_dataset_uri
 - DistillationSpec.tuned_teacher_model_source
 - DistillationSpec.validation_dataset_uri
- DistillationSpecDict
 - DistillationSpecDict.base_teacher_model
 - DistillationSpecDict.hyper_parameters
 - DistillationSpecDict.pipeline_root_directory
 - DistillationSpecDict.student_model
 - DistillationSpecDict.training_dataset_uri
 - DistillationSpecDict.tuned_teacher_model_source
 - DistillationSpecDict.validation_dataset_uri
- DownloadFileConfig
 - DownloadFileConfig.http_options

Skip to content `fileConfig.http_options`
`.eConfigDict`

- DownloadFileConfigDict.http_options

- DynamicRetrievalConfig
 - DynamicRetrievalConfig.dynamic_threshold
 - DynamicRetrievalConfig.mode
 - DynamicRetrievalConfigDict
 - DynamicRetrievalConfigDict.dynamic_threshold
 - DynamicRetrievalConfigDict.mode
 - DynamicRetrievalConfigMode
 - DynamicRetrievalConfigMode.MODE_DYNAMIC
 - DynamicRetrievalConfigMode.MODE_UNSPECIFIED
 - EditImageConfig
 - EditImageConfig.aspect_ratio
 - EditImageConfig.base_steps
 - EditImageConfig.edit_mode
 - EditImageConfig.guidance_scale
 - EditImageConfig.http_options
 - EditImageConfig.include_rai_reason
 - EditImageConfig.include_safety_attributes
 - EditImageConfig.language
 - EditImageConfig.negative_prompt
 - EditImageConfig.number_of_images
 - EditImageConfig.output_compression_quality
 - EditImageConfig.output_gcs_uri
 - EditImageConfig.output_mime_type
 - EditImageConfig.person_generation
 - EditImageConfig.safety_filter_level
 - EditImageConfig.seed
 - EditImageConfigDict
 - EditImageConfigDict.aspect_ratio
 - EditImageConfigDict.base_steps
 - EditImageConfigDict.edit_mode
 - EditImageConfigDict.guidance_scale
 - EditImageConfigDict.http_options
 - EditImageConfigDict.include_rai_reason
 - EditImageConfigDict.include_safety_attributes
 - EditImageConfigDict.language
 - EditImageConfigDict.negative_prompt
 - EditImageConfigDict.number_of_images
 - EditImageConfigDict.output_compression_quality
 - EditImageConfigDict.output_gcs_uri
 - EditImageConfigDict.output_mime_type
 - EditImageConfigDict.person_generation
 - EditImageConfigDict.safety_filter_level
- Skip to content `:ConfigDict.seed`
 `.response`
- EditImageResponse.generated_images

- `EditImageResponseDict`
 - `EditImageResponseDict.generated_images`
- `EditMode`
 - `EditMode.EDIT_MODE_BGSWAP`
 - `EditMode.EDIT_MODE_CONTROLLED_EDITING`
 - `EditMode.EDIT_MODE_DEFAULT`
 - `EditMode.EDIT_MODE_INPAINT_INSERTION`
 - `EditMode.EDIT_MODE_INPAINT_REMOVAL`
 - `EditMode.EDIT_MODE_OUTPAINT`
 - `EditMode.EDIT_MODE_PRODUCT_IMAGE`
 - `EditMode.EDIT_MODE_STYLE`
- `EmbedContentConfig`
 - `EmbedContentConfig.auto_truncate`
 - `EmbedContentConfig.http_options`
 - `EmbedContentConfig.mime_type`
 - `EmbedContentConfig.output_dimensionality`
 - `EmbedContentConfig.task_type`
 - `EmbedContentConfig.title`
- `EmbedContentConfigDict`
 - `EmbedContentConfigDict.auto_truncate`
 - `EmbedContentConfigDict.http_options`
 - `EmbedContentConfigDict.mime_type`
 - `EmbedContentConfigDict.output_dimensionality`
 - `EmbedContentConfigDict.task_type`
 - `EmbedContentConfigDict.title`
- `EmbedContentMetadata`
 - `EmbedContentMetadata.billable_character_count`
- `EmbedContentMetadataDict`
 - `EmbedContentMetadataDict.billable_character_count`
- `EmbedContentResponse`
 - `EmbedContentResponse.embeddings`
 - `EmbedContentResponse.metadata`
- `EmbedContentResponseDict`
 - `EmbedContentResponseDict.embeddings`
 - `EmbedContentResponseDict.metadata`
- `EncryptionSpec`
 - `EncryptionSpec.kms_key_name`
- `EncryptionSpecDict`
 - `EncryptionSpecDict.kms_key_name`
- `Endpoint`
 - `Endpoint.deployed_model_id`
 - `Endpoint.name`

Skip to content `t`
 `dict.deployed_model_id`

- `EndpointDict.name`

- ExecutableCode
 - ExecutableCode.code
 - ExecutableCode.language
- ExecutableCodeDict
 - ExecutableCodeDict.code
 - ExecutableCodeDict.language
- FetchPredictOperationConfig
 - FetchPredictOperationConfig.http_options
- FetchPredictOperationConfigDict
 - FetchPredictOperationConfigDict.http_options
- File
 - File.create_time
 - File.display_name
 - File.download_uri
 - File.error
 - File.expiration_time
 - File.mime_type
 - File.name
 - File.sha256_hash
 - File.size_bytes
 - File.source
 - File.state
 - File.update_time
 - File.uri
 - File.video_metadata
- FileData
 - FileData.file_uri
 - FileData.mime_type
- FileDataDict
 - FileDataDict.file_uri
 - FileDataDict.mime_type

[Skip to content](#)

- `FileDialog`
 - `FileDialog.create_time`
 - `FileDialog.display_name`
 - `FileDialog.download_uri`
 - `FileDialog.error`
 - `FileDialog.expiration_time`
 - `FileDialog.mime_type`
 - `FileDialog.name`
 - `FileDialog.sha256_hash`
 - `FileDialog.size_bytes`
 - `FileDialog.source`
 - `FileDialog.state`
 - `FileDialog.update_time`
 - `FileDialog.uri`
 - `FileDialog.video_metadata`
- `FileSource`
 - `FileSource.GENERATED`
 - `FileSource.SOURCE_UNSPECIFIED`
 - `FileSource.UPLOADED`
- `FileState`
 - `FileState.ACTIVE`
 - `FileState.FAILED`
 - `FileState.PROCESSING`
 - `FileState.STATE_UNSPECIFIED`
- `FileStatus`
 - `FileStatus.code`
 - `FileStatus.details`
 - `FileStatus.message`
- `FileStatusDict`
 - `FileStatusDict.code`
 - `FileStatusDict.details`
 - `FileStatusDict.message`
- `FinishReason`
 - `FinishReason.BLOCKLIST`
 - `FinishReason.FINISH_REASON_UNSPECIFIED`
 - `FinishReason.MALFORMED_FUNCTION_CALL`
 - `FinishReason.MAX_TOKENS`
 - `FinishReason.OTHER`
 - `FinishReason.PROHIBITED_CONTENT`
 - `FinishReason.RECITATION`
 - `FinishReason.SAFETY`
 - `FinishReason.SPII`

Skip to content [JSON STOP](#)

- FunctionCall
 - FunctionCall.args
 - FunctionCall.id
 - FunctionCall.name
- FunctionCallDict
 - FunctionCallDict.args
 - FunctionCallDict.id
 - FunctionCallDict.name
- FunctionCallingConfig
 - FunctionCallingConfig.allowed_function_names
 - FunctionCallingConfig.mode
- FunctionCallingConfigDict
 - FunctionCallingConfigDict.allowed_function_names
 - FunctionCallingConfigDict.mode
- FunctionCallingConfigMode
 - FunctionCallingConfigMode.ANY
 - FunctionCallingConfigMode.AUTO
 - FunctionCallingConfigMode.MODE_UNSPECIFIED
 - FunctionCallingConfigMode.NONE
- FunctionDeclaration
 - FunctionDeclaration.description
 - FunctionDeclaration.name
 - FunctionDeclaration.parameters
 - FunctionDeclaration.response
 - FunctionDeclaration.from_callable()
 - FunctionDeclaration.from_callable_with_api_option()
- FunctionDeclarationDict
 - FunctionDeclarationDict.description
 - FunctionDeclarationDict.name
 - FunctionDeclarationDict.parameters
 - FunctionDeclarationDict.response
- FunctionResponse
 - FunctionResponse.id
 - FunctionResponse.name
 - FunctionResponse.response
- FunctionResponseDict
 - FunctionResponseDict.id
 - FunctionResponseDict.name
 - FunctionResponseDict.response

[Skip to content](#)

- GenerateContentConfig
 - GenerateContentConfig.audio_timestamp
 - GenerateContentConfig.automatic_function_calling
 - GenerateContentConfig.cached_content
 - GenerateContentConfig.candidate_count
 - GenerateContentConfig.frequency_penalty
 - GenerateContentConfig.http_options
 - GenerateContentConfig.labels
 - GenerateContentConfig.logprobs
 - GenerateContentConfig.max_output_tokens
 - GenerateContentConfig.media_resolution
 - GenerateContentConfig.presence_penalty
 - GenerateContentConfig.response_logprobs
 - GenerateContentConfig.response_mime_type
 - GenerateContentConfig.response_modalities
 - GenerateContentConfig.response_schema
 - GenerateContentConfig.routing_config
 - GenerateContentConfig.safety_settings
 - GenerateContentConfig.seed
 - GenerateContentConfig.speech_config
 - GenerateContentConfig.stop_sequences
 - GenerateContentConfig.system_instruction
 - GenerateContentConfig.temperature
 - GenerateContentConfig.thinking_config
 - GenerateContentConfig.tool_config
 - GenerateContentConfig.tools
 - GenerateContentConfig.top_k
 - GenerateContentConfig.top_p

[Skip to content](#)

- GenerateContentConfigDict
 - GenerateContentConfigDict.audio_timestamp
 - GenerateContentConfigDict.automatic_function_calling
 - GenerateContentConfigDict.cached_content
 - GenerateContentConfigDict.candidate_count
 - GenerateContentConfigDict.frequency_penalty
 - GenerateContentConfigDict.http_options
 - GenerateContentConfigDict.labels
 - GenerateContentConfigDict.logprobs
 - GenerateContentConfigDict.max_output_tokens
 - GenerateContentConfigDict.media_resolution
 - GenerateContentConfigDict.presence_penalty
 - GenerateContentConfigDict.response_logprobs
 - GenerateContentConfigDict.response_mime_type
 - GenerateContentConfigDict.response_modalities
 - GenerateContentConfigDict.response_schema
 - GenerateContentConfigDict.routing_config
 - GenerateContentConfigDict.safety_settings
 - GenerateContentConfigDict.seed
 - GenerateContentConfigDict.speech_config
 - GenerateContentConfigDict.stop_sequences
 - GenerateContentConfigDict.system_instruction
 - GenerateContentConfigDict.temperature
 - GenerateContentConfigDict.thinking_config
 - GenerateContentConfigDict.tool_config
 - GenerateContentConfigDict.tools
 - GenerateContentConfigDict.top_k
 - GenerateContentConfigDict.top_p
- GenerateContentResponse
 - GenerateContentResponse.automatic_function_calling_history
 - GenerateContentResponse.candidates
 - GenerateContentResponse.create_time
 - GenerateContentResponse.model_version
 - GenerateContentResponse.parsed
 - GenerateContentResponse.prompt_feedback
 - GenerateContentResponse.response_id
 - GenerateContentResponse.usage_metadata
 - GenerateContentResponse.code_execution_result
 - GenerateContentResponse.executable_code
 - GenerateContentResponse.function_calls
 - GenerateContentResponse.text

[Skip to content](#)

- GenerateContentResponseDict
 - GenerateContentResponseDict.candidates
 - GenerateContentResponseDict.create_time
 - GenerateContentResponseDict.model_version
 - GenerateContentResponseDict.prompt_feedback
 - GenerateContentResponseDict.response_id
 - GenerateContentResponseDict.usage_metadata
- GenerateContentResponsePromptFeedback
 - GenerateContentResponsePromptFeedback.block_reason
 - GenerateContentResponsePromptFeedback.block_reason_message
 - GenerateContentResponsePromptFeedback.safety_ratings
- GenerateContentResponsePromptFeedbackDict
 - GenerateContentResponsePromptFeedbackDict.block_reason
 - GenerateContentResponsePromptFeedbackDict.block_reason_message
 - GenerateContentResponsePromptFeedbackDict.safety_ratings
- GenerateContentResponseUsageMetadata
 - GenerateContentResponseUsageMetadata.cached_content_token_count
 - GenerateContentResponseUsageMetadata.candidates_token_count
 - GenerateContentResponseUsageMetadata.prompt_token_count
 - GenerateContentResponseUsageMetadata.total_token_count
- GenerateContentResponseUsageMetadataDict
 - GenerateContentResponseUsageMetadataDict.cached_content_token_count
 - GenerateContentResponseUsageMetadataDict.candidates_token_count
 - GenerateContentResponseUsageMetadataDict.prompt_token_count
 - GenerateContentResponseUsageMetadataDict.total_token_count
- GenerateImagesConfig
 - GenerateImagesConfig.add_watermark
 - GenerateImagesConfig.aspect_ratio
 - GenerateImagesConfig.enhance_prompt
 - GenerateImagesConfig.guidance_scale
 - GenerateImagesConfig.http_options
 - GenerateImagesConfig.include_rai_reason
 - GenerateImagesConfig.include_safety_attributes
 - GenerateImagesConfig.language
 - GenerateImagesConfig.negative_prompt
 - GenerateImagesConfig.number_of_images
 - GenerateImagesConfig.output_compression_quality
 - GenerateImagesConfig.output_gcs_uri
 - GenerateImagesConfig.output_mime_type
 - GenerateImagesConfig.person_generation
 - GenerateImagesConfig.safety_filter_level
 - GenerateImagesConfig.seed

[Skip to content](#)

- `GenerateImagesConfigDict`
 - `GenerateImagesConfigDict.add_watermark`
 - `GenerateImagesConfigDict.aspect_ratio`
 - `GenerateImagesConfigDict.enhance_prompt`
 - `GenerateImagesConfigDict.guidance_scale`
 - `GenerateImagesConfigDict.http_options`
 - `GenerateImagesConfigDict.include_rai_reason`
 - `GenerateImagesConfigDict.include_safety_attributes`
 - `GenerateImagesConfigDict.language`
 - `GenerateImagesConfigDict.negative_prompt`
 - `GenerateImagesConfigDict.number_of_images`
 - `GenerateImagesConfigDict.output_compression_quality`
 - `GenerateImagesConfigDict.output_gcs_uri`
 - `GenerateImagesConfigDict.output_mime_type`
 - `GenerateImagesConfigDict.person_generation`
 - `GenerateImagesConfigDict.safety_filter_level`
 - `GenerateImagesConfigDict.seed`
- `GenerateImagesResponse`
 - `GenerateImagesResponse.generated_images`
- `GenerateImagesResponseDict`
 - `GenerateImagesResponseDict.generated_images`
- `GenerateVideosConfig`
 - `GenerateVideosConfig.aspect_ratio`
 - `GenerateVideosConfig.duration_seconds`
 - `GenerateVideosConfig.enhance_prompt`
 - `GenerateVideosConfig.fps`
 - `GenerateVideosConfig.http_options`
 - `GenerateVideosConfig.negative_prompt`
 - `GenerateVideosConfig.number_of_videos`
 - `GenerateVideosConfig.output_gcs_uri`
 - `GenerateVideosConfig.person_generation`
 - `GenerateVideosConfig.pubsub_topic`
 - `GenerateVideosConfig.resolution`
 - `GenerateVideosConfig.seed`

[Skip to content](#)

- GenerateVideosConfigDict
 - GenerateVideosConfigDict.aspect_ratio
 - GenerateVideosConfigDict.duration_seconds
 - GenerateVideosConfigDict.enhance_prompt
 - GenerateVideosConfigDict.fps
 - GenerateVideosConfigDict.http_options
 - GenerateVideosConfigDict.negative_prompt
 - GenerateVideosConfigDict.number_of_videos
 - GenerateVideosConfigDict.output_gcs_uri
 - GenerateVideosConfigDict.person_generation
 - GenerateVideosConfigDict.pubsub_topic
 - GenerateVideosConfigDict.resolution
 - GenerateVideosConfigDict.seed
 - GenerateVideosOperation
 - GenerateVideosOperation.done
 - GenerateVideosOperation.error
 - GenerateVideosOperation.metadata
 - GenerateVideosOperation.name
 - GenerateVideosOperation.response
 - GenerateVideosOperation.result
 - GenerateVideosOperationDict
 - GenerateVideosOperationDict.done
 - GenerateVideosOperationDict.error
 - GenerateVideosOperationDict.metadata
 - GenerateVideosOperationDict.name
 - GenerateVideosOperationDict.response
 - GenerateVideosOperationDict.result
 - GenerateVideosResponse
 - GenerateVideosResponse.generated_videos
 - GenerateVideosResponse.rai_media_filtered_count
 - GenerateVideosResponse.rai_media_filtered_reasons
 - GenerateVideosResponseDict
 - GenerateVideosResponseDict.generated_videos
 - GenerateVideosResponseDict.rai_media_filtered_count
 - GenerateVideosResponseDict.rai_media_filtered_reasons
 - GeneratedImage
 - GeneratedImage.enhanced_prompt
 - GeneratedImage.image
 - GeneratedImage.rai_filtered_reason
 - GeneratedImage.safety_attributes
 - GeneratedImageDict
 - GeneratedImageDict.enhanced_prompt
- Skip to content `iImageDict.image`
 `jImageDict.rai_filtered_reason`
 - GeneratedImageDict.safety_attributes

- GeneratedVideo
 - GeneratedVideo.video
- GeneratedVideoDict
 - GeneratedVideoDict.video
- GenerationConfig
 - GenerationConfig.audio_timestamp
 - GenerationConfig.candidate_count
 - GenerationConfig.frequency_penalty
 - GenerationConfig.logprobs
 - GenerationConfig.max_output_tokens
 - GenerationConfig.presence_penalty
 - GenerationConfig.response_logprobs
 - GenerationConfig.response_mime_type
 - GenerationConfig.response_schema
 - GenerationConfig.routing_config
 - GenerationConfig.seed
 - GenerationConfig.stop_sequences
 - GenerationConfig.temperature
 - GenerationConfig.top_k
 - GenerationConfig.top_p
- GenerationConfigDict
 - GenerationConfigDict.audio_timestamp
 - GenerationConfigDict.candidate_count
 - GenerationConfigDict.frequency_penalty
 - GenerationConfigDict.logprobs
 - GenerationConfigDict.max_output_tokens
 - GenerationConfigDict.presence_penalty
 - GenerationConfigDict.response_logprobs
 - GenerationConfigDict.response_mime_type
 - GenerationConfigDict.response_schema
 - GenerationConfigDict.routing_config
 - GenerationConfigDict.seed
 - GenerationConfigDict.stop_sequences
 - GenerationConfigDict.temperature
 - GenerationConfigDict.top_k
 - GenerationConfigDict.top_p
- GenerationConfigRoutingConfig
 - GenerationConfigRoutingConfig.auto_mode
 - GenerationConfigRoutingConfig.manual_mode
- GenerationConfigRoutingConfigAutoRoutingMode
 - GenerationConfigRoutingConfigAutoRoutingMode.model_routing_preference
- GenerationConfigRoutingConfigAutoRoutingModeDict
 - GenerationConfigRoutingConfigAutoRoutingModeDict.model_routing_preference

Skip to content

- GenerationConfigRoutingConfigDict
 - GenerationConfigRoutingConfigDict.auto_mode
 - GenerationConfigRoutingConfigDict.manual_mode
- GenerationConfigRoutingConfigManualRoutingMode
 - GenerationConfigRoutingConfigManualRoutingMode.model_name
- GenerationConfigRoutingConfigManualRoutingModeDict
 - GenerationConfigRoutingConfigManualRoutingModeDict.model_name
- GetBatchJobConfig
 - GetBatchJobConfig.http_options
- GetBatchJobConfigDict
 - GetBatchJobConfigDict.http_options
- GetCachedContentConfig
 - GetCachedContentConfig.http_options
- GetCachedContentConfigDict
 - GetCachedContentConfigDict.http_options
- GetFileConfig
 - GetFileConfig.http_options
- GetFileConfigDict
 - GetFileConfigDict.http_options
- GetModelConfig
 - GetModelConfig.http_options
- GetModelConfigDict
 - GetModelConfigDict.http_options
- GetOperationConfig
 - GetOperationConfig.http_options
- GetOperationConfigDict
 - GetOperationConfigDict.http_options
- GetTuningJobConfig
 - GetTuningJobConfig.http_options
- GetTuningJobConfigDict
 - GetTuningJobConfigDict.http_options
- GoogleRpcStatus
 - GoogleRpcStatus.code
 - GoogleRpcStatus.details
 - GoogleRpcStatus.message
- GoogleRpcStatusDict
 - GoogleRpcStatusDict.code
 - GoogleRpcStatusDict.details
 - GoogleRpcStatusDict.message
- GoogleSearch
- GoogleSearchDict
- GoogleSearchRetrieval

Skip to content `.archRetrieval.dynamic_retrieval_config`

`.hRetrievalDict`

`▪ GoogleSearchRetrievalDict.dynamic_retrieval_config`

- GoogleTypeDate
 - `GoogleTypeDate.day`
 - `GoogleTypeDate.month`
 - `GoogleTypeDate.year`
- GoogleTypeDateDict
 - `GoogleTypeDateDict.day`
 - `GoogleTypeDateDict.month`
 - `GoogleTypeDateDict.year`
- GroundingChunk
 - `GroundingChunk.retrieved_context`
 - `GroundingChunk.web`
- GroundingChunkDict
 - `GroundingChunkDict.retrieved_context`
 - `GroundingChunkDict.web`
- GroundingChunkRetrievedContext
 - `GroundingChunkRetrievedContext.text`
 - `GroundingChunkRetrievedContext.title`
 - `GroundingChunkRetrievedContext.uri`
- GroundingChunkRetrievedContextDict
 - `GroundingChunkRetrievedContextDict.text`
 - `GroundingChunkRetrievedContextDict.title`
 - `GroundingChunkRetrievedContextDict.uri`
- GroundingChunkWeb
 - `GroundingChunkWeb.title`
 - `GroundingChunkWeb.uri`
- GroundingChunkWebDict
 - `GroundingChunkWebDict.title`
 - `GroundingChunkWebDict.uri`
- GroundingMetadata
 - `GroundingMetadata.grounding_chunks`
 - `GroundingMetadata.grounding_supports`
 - `GroundingMetadata.retrieval_metadata`
 - `GroundingMetadata.retrieval_queries`
 - `GroundingMetadata.search_entry_point`
 - `GroundingMetadata.web_search_queries`
- GroundingMetadataDict
 - `GroundingMetadataDict.grounding_chunks`
 - `GroundingMetadataDict.grounding_supports`
 - `GroundingMetadataDict.retrieval_metadata`
 - `GroundingMetadataDict.retrieval_queries`
 - `GroundingMetadataDict.search_entry_point`
 - `GroundingMetadataDict.web_search_queries`

[Skip to content](#)

- GroundingSupport
 - `GroundingSupport.confidence_scores`
 - `GroundingSupport.grounding_chunk_indices`
 - `GroundingSupport.segment`
- GroundingSupportDict
 - `GroundingSupportDict.confidence_scores`
 - `GroundingSupportDict.grounding_chunk_indices`
 - `GroundingSupportDict.segment`
- HarmBlockMethod
 - `HarmBlockMethod.HARM_BLOCK_METHOD_UNSPECIFIED`
 - `HarmBlockMethod.PROBABILITY`
 - `HarmBlockMethod.SEVERITY`
- HarmBlockThreshold
 - `HarmBlockThreshold.BLOCK_LOW_AND ABOVE`
 - `HarmBlockThreshold.BLOCK_MEDIUM_AND ABOVE`
 - `HarmBlockThreshold.BLOCK_NONE`
 - `HarmBlockThreshold.BLOCK_ONLY_HIGH`
 - `HarmBlockThreshold.HARM_BLOCK_THRESHOLD_UNSPECIFIED`
 - `HarmBlockThreshold.OFF`
- HarmCategory
 - `HarmCategory.HARM_CATEGORY_CIVIC_INTEGRITY`
 - `HarmCategory.HARM_CATEGORY_DANGEROUS_CONTENT`
 - `HarmCategory.HARM_CATEGORY_HARASSMENT`
 - `HarmCategory.HARM_CATEGORY_HATE_SPEECH`
 - `HarmCategory.HARM_CATEGORY_SEXUALLY_EXPLICIT`
 - `HarmCategory.HARM_CATEGORY_UNSPECIFIED`
- HarmProbability
 - `HarmProbability.HARM_PROBABILITY_UNSPECIFIED`
 - `HarmProbability.HIGH`
 - `HarmProbability.LOW`
 - `HarmProbability.MEDIUM`
 - `HarmProbability.NEGLIGIBLE`
- HarmSeverity
 - `HarmSeverity.HARM_SEVERITY_HIGH`
 - `HarmSeverity.HARM_SEVERITY_LOW`
 - `HarmSeverity.HARM_SEVERITY_MEDIUM`
 - `HarmSeverity.HARM_SEVERITY_NEGLIGIBLE`
 - `HarmSeverity.HARM_SEVERITY_UNSPECIFIED`
- HttpOptions
 - `HttpOptions.api_version`
 - `HttpOptions.base_url`
 - `HttpOptions.headers`

Skip to content `>ns.timeout`

- `HttpOptionsDict`
 - `HttpOptionsDict.api_version`
 - `HttpOptionsDict.base_url`
 - `HttpOptionsDict.headers`
 - `HttpOptionsDict.timeout`
- `Image`
 - `Image.gcs_uri`
 - `Image.image_bytes`
 - `Image.mime_type`
 - `Image.from_file()`
 - `Image.model_post_init()`
 - `Image.save()`
 - `Image.show()`
- `ImageDict`
 - `ImageDict.gcs_uri`
 - `ImageDict.image_bytes`
 - `ImageDict.mime_type`
- `ImagePromptLanguage`
 - `ImagePromptLanguage.auto`
 - `ImagePromptLanguage.en`
 - `ImagePromptLanguage.hi`
 - `ImagePromptLanguage.ja`
 - `ImagePromptLanguage.ko`
- `JobError`
 - `JobError.code`
 - `JobError.details`
 - `JobError.message`
- `JobErrorDict`
 - `JobErrorDict.code`
 - `JobErrorDict.details`
 - `JobErrorDict.message`
- `JobState`
 - `JobState.JOB_STATE_CANCELLED`
 - `JobState.JOB_STATE_CANCELLING`
 - `JobState.JOB_STATE_EXPIRED`
 - `JobState.JOB_STATE_FAILED`
 - `JobState.JOB_STATE_PARTIALLY_SUCCEEDED`
 - `JobState.JOB_STATE_PAUSED`
 - `JobState.JOB_STATE_PENDING`
 - `JobState.JOB_STATE_QUEUED`
 - `JobState.JOB_STATE_RUNNING`
 - `JobState.JOB_STATE_SUCCEEDED`
 - `JobState.JOB_STATE_UNSPECIFIED`
 - `JobState.JOB_STATE_UPDATING`

Skip to content .`JOB_STATE_UNSPECIFIED`
.`JOB_STATE_UPDATING`

- Language
 - Language.LANGUAGE_UNSPECIFIED
 - Language.PYTHON
- ListBatchJobsConfig
 - ListBatchJobsConfig.filter
 - ListBatchJobsConfig.http_options
 - ListBatchJobsConfig.page_size
 - ListBatchJobsConfig.page_token
- ListBatchJobsConfigDict
 - ListBatchJobsConfigDict.filter
 - ListBatchJobsConfigDict.http_options
 - ListBatchJobsConfigDict.page_size
 - ListBatchJobsConfigDict.page_token
- ListBatchJobsResponse
 - ListBatchJobsResponse.batch_jobs
 - ListBatchJobsResponse.next_page_token
- ListBatchJobsResponseDict
 - ListBatchJobsResponseDict.batch_jobs
 - ListBatchJobsResponseDict.next_page_token
- ListCachedContentsConfig
 - ListCachedContentsConfig.http_options
 - ListCachedContentsConfig.page_size
 - ListCachedContentsConfig.page_token
- ListCachedContentsConfigDict
 - ListCachedContentsConfigDict.http_options
 - ListCachedContentsConfigDict.page_size
 - ListCachedContentsConfigDict.page_token
- ListCachedContentsResponse
 - ListCachedContentsResponse.cached_contents
 - ListCachedContentsResponse.next_page_token
- ListCachedContentsResponseDict
 - ListCachedContentsResponseDict.cached_contents
 - ListCachedContentsResponseDict.next_page_token
- ListFilesConfig
 - ListFilesConfig.http_options
 - ListFilesConfig.page_size
 - ListFilesConfig.page_token
- ListFilesConfigDict
 - ListFilesConfigDict.http_options
 - ListFilesConfigDict.page_size
 - ListFilesConfigDict.page_token
- ListFilesResponse
 - Response.files
 - Response.next_page_token

Skip to content ↵Response.files
 ↳Response.next_page_token

- `ListFilesResponseDict`
 - `ListFilesResponseDict.files`
 - `ListFilesResponseDict.next_page_token`
- `ListModelsConfig`
 - `ListModelsConfig.filter`
 - `ListModelsConfig.http_options`
 - `ListModelsConfig.page_size`
 - `ListModelsConfig.page_token`
 - `ListModelsConfig.query_base`
- `ListModelsConfigDict`
 - `ListModelsConfigDict.filter`
 - `ListModelsConfigDict.http_options`
 - `ListModelsConfigDict.page_size`
 - `ListModelsConfigDict.page_token`
 - `ListModelsConfigDict.query_base`
- `ListModelsResponse`
 - `ListModelsResponse.models`
 - `ListModelsResponse.next_page_token`
- `ListModelsResponseDict`
 - `ListModelsResponseDict.models`
 - `ListModelsResponseDict.next_page_token`
- `ListTuningJobsConfig`
 - `ListTuningJobsConfig.filter`
 - `ListTuningJobsConfig.http_options`
 - `ListTuningJobsConfig.page_size`
 - `ListTuningJobsConfig.page_token`
- `ListTuningJobsConfigDict`
 - `ListTuningJobsConfigDict.filter`
 - `ListTuningJobsConfigDict.http_options`
 - `ListTuningJobsConfigDict.page_size`
 - `ListTuningJobsConfigDict.page_token`
- `ListTuningJobsResponse`
 - `ListTuningJobsResponse.next_page_token`
 - `ListTuningJobsResponse.tuning_jobs`
- `ListTuningJobsResponseDict`
 - `ListTuningJobsResponseDict.next_page_token`
 - `ListTuningJobsResponseDict.tuning_jobs`
- `LiveClientContent`
 - `LiveClientContent.turn_complete`
 - `LiveClientContent.turns`
- `LiveClientContentDict`
 - `LiveClientContentDict.turn_complete`

Skip to content `\tContentDict.turns`

- `LiveClientMessage`
 - `LiveClientMessage.client_content`
 - `LiveClientMessage.realtime_input`
 - `LiveClientMessage.setup`
 - `LiveClientMessage.tool_response`
 - `LiveClientMessageDict`
 - `LiveClientMessageDict.client_content`
 - `LiveClientMessageDict.realtime_input`
 - `LiveClientMessageDict.setup`
 - `LiveClientMessageDict.tool_response`
 - `LiveClientRealtimeInput`
 - `LiveClientRealtimeInput.media_chunks`
 - `LiveClientRealtimeInputDict`
 - `LiveClientRealtimeInputDict.media_chunks`
 - `LiveClientSetup`
 - `LiveClientSetup.generation_config`
 - `LiveClientSetup.model`
 - `LiveClientSetup.system_instruction`
 - `LiveClientSetup.tools`
 - `LiveClientSetupDict`
 - `LiveClientSetupDict.generation_config`
 - `LiveClientSetupDict.model`
 - `LiveClientSetupDict.system_instruction`
 - `LiveClientSetupDict.tools`
 - `LiveClientToolResponse`
 - `LiveClientToolResponse.function_responses`
 - `LiveClientToolResponseDict`
 - `LiveClientToolResponseDict.function_responses`
 - `LiveConnectConfig`
 - `LiveConnectConfig.generation_config`
 - `LiveConnectConfig.response_modalities`
 - `LiveConnectConfig.speech_config`
 - `LiveConnectConfig.system_instruction`
 - `LiveConnectConfig.tools`
 - `LiveConnectConfigDict`
 - `LiveConnectConfigDict.generation_config`
 - `LiveConnectConfigDict.response_modalities`
 - `LiveConnectConfigDict.speech_config`
 - `LiveConnectConfigDict.system_instruction`
 - `LiveConnectConfigDict.tools`
 - `LiveServerContent`
 - `LiveServerContent.interrupted`
- Skip to content `>rContent.model_turn`
`>rContent.turn_complete`

- `LiveServerContentDict`
 - `LiveServerContentDict.interrupted`
 - `LiveServerContentDict.model_turn`
 - `LiveServerContentDict.turn_complete`
- `LiveServerMessage`
 - `LiveServerMessage.server_content`
 - `LiveServerMessage.setup_complete`
 - `LiveServerMessage.tool_call`
 - `LiveServerMessage.tool_call_cancellation`
 - `LiveServerMessage.data`
 - `LiveServerMessage.text`
- `LiveServerMessageDict`
 - `LiveServerMessageDict.server_content`
 - `LiveServerMessageDict.setup_complete`
 - `LiveServerMessageDict.tool_call`
 - `LiveServerMessageDict.tool_call_cancellation`
- `LiveServerSetupComplete`
- `LiveServerSetupCompleteDict`
- `LiveServerToolCall`
 - `LiveServerToolCall.function_calls`
- `LiveServerToolCallCancellation`
 - `LiveServerToolCallCancellation.ids`
- `LiveServerToolCallCancellationDict`
 - `LiveServerToolCallCancellationDict.ids`
- `LiveServerToolCallDict`
 - `LiveServerToolCallDict.function_calls`
- `LogprobsResult`
 - `LogprobsResult.chosen_candidates`
 - `LogprobsResult.top_candidates`
- `LogprobsResultCandidate`
 - `LogprobsResultCandidate.log_probability`
 - `LogprobsResultCandidate.token`
 - `LogprobsResultCandidate.token_id`
- `LogprobsResultCandidateDict`
 - `LogprobsResultCandidateDict.log_probability`
 - `LogprobsResultCandidateDict.token`
 - `LogprobsResultCandidateDict.token_id`
- `LogprobsResultDict`
 - `LogprobsResultDict.chosen_candidates`
 - `LogprobsResultDict.top_candidates`
- `LogprobsResultTopCandidates`
 - `LogprobsResultTopCandidates.candidates`

Skip to content `ultTopCandidatesDict`
`ResultTopCandidatesDict.candidates`

- MaskReferenceConfig
 - MaskReferenceConfig.mask_dilation
 - MaskReferenceConfig.mask_mode
 - MaskReferenceConfig.segmentation_classes
- MaskReferenceConfigDict
 - MaskReferenceConfigDict.mask_dilation
 - MaskReferenceConfigDict.mask_mode
 - MaskReferenceConfigDict.segmentation_classes
- MaskReferenceImage
 - MaskReferenceImage.config
 - MaskReferenceImage.mask_image_config
 - MaskReferenceImage.reference_id
 - MaskReferenceImage.reference_image
 - MaskReferenceImage.reference_type
- MaskReferenceImageDict
 - MaskReferenceImageDict.config
 - MaskReferenceImageDict.reference_id
 - MaskReferenceImageDict.reference_image
 - MaskReferenceImageDict.reference_type
- MaskReferenceMode
 - MaskReferenceMode.MASK_MODE_BACKGROUND
 - MaskReferenceMode.MASK_MODE_DEFAULT
 - MaskReferenceMode.MASK_MODE_FOREGROUND
 - MaskReferenceMode.MASK_MODE_SEMANTIC
 - MaskReferenceMode.MASK_MODE_USER_PROVIDED
- MediaResolution
 - MediaResolution.MEDIA_RESOLUTION_HIGH
 - MediaResolution.MEDIA_RESOLUTION_LOW
 - MediaResolution.MEDIA_RESOLUTION_MEDIUM
 - MediaResolution.MEDIA_RESOLUTION_UNSPECIFIED
- Modality
 - Modality.AUDIO
 - Modality.IMAGE
 - Modality.MODALITY_UNSPECIFIED
 - Modality.TEXT
- Mode
 - Mode.MODE_DYNAMIC
 - Mode.MODE_UNSPECIFIED

[Skip to content](#)

- Model
 - `Model.description`
 - `Model.display_name`
 - `Model.endpoints`
 - `Model.input_token_limit`
 - `Model.labels`
 - `Model.name`
 - `Model.output_token_limit`
 - `Model.supported_actions`
 - `Model.tuned_model_info`
 - `Model.version`
- ModelContent
 - `ModelContent.parts`
 - `ModelContent.role`
- ModelDict
 - `ModelDict.description`
 - `ModelDict.display_name`
 - `ModelDict.endpoints`
 - `ModelDict.input_token_limit`
 - `ModelDict.labels`
 - `ModelDict.name`
 - `ModelDict.output_token_limit`
 - `ModelDict.supported_actions`
 - `ModelDict.tuned_model_info`
 - `ModelDict.version`
- Operation
 - `Operation.done`
 - `Operation.error`
 - `Operation.metadata`
 - `Operation.name`
 - `Operation.response`
- OperationDict
 - `OperationDict.done`
 - `OperationDict.error`
 - `OperationDict.metadata`
 - `OperationDict.name`
 - `OperationDict.response`
- Outcome
 - `Outcome.OUTCOME_DEADLINE_EXCEEDED`
 - `Outcome.OUTCOME_FAILED`
 - `Outcome.OUTCOME_OK`
 - `Outcome.OUTCOME_UNSPECIFIED`

[Skip to content](#)

- Part
 - `Part.code_execution_result`
 - `Part.executable_code`
 - `Part.file_data`
 - `Part.function_call`
 - `Part.function_response`
 - `Part.inline_data`
 - `Part.text`
 - `Part.thought`
 - `Part.video_metadata`
 - `Part.from_bytes()`
 - `Part.from_code_execution_result()`
 - `Part.from_executable_code()`
 - `Part.from_function_call()`
 - `Part.from_function_response()`
 - `Part.from_text()`
 - `Part.from_uri()`
 - `Part.from_video_metadata()`
- PartDict
 - `PartDict.code_execution_result`
 - `PartDict.executable_code`
 - `PartDict.file_data`
 - `PartDict.function_call`
 - `PartDict.function_response`
 - `PartDict.inline_data`
 - `PartDict.text`
 - `PartDict.thought`
 - `PartDict.video_metadata`
- PartnerModelTuningSpec
 - `PartnerModelTuningSpec.hyper_parameters`
 - `PartnerModelTuningSpec.training_dataset_uri`
 - `PartnerModelTuningSpec.validation_dataset_uri`
- PartnerModelTuningSpecDict
 - `PartnerModelTuningSpecDict.hyper_parameters`
 - `PartnerModelTuningSpecDict.training_dataset_uri`
 - `PartnerModelTuningSpecDict.validation_dataset_uri`
- PersonGeneration
 - `PersonGeneration.ALLOW_ADULT`
 - `PersonGeneration.ALLOW_ALL`
 - `PersonGeneration.DONT_ALLOW`
- PrebuiltVoiceConfig
 - `PrebuiltVoiceConfig.voice_name`

Skip to content `ceConfigDict`
`/oiceConfigDict.voice_name`

- RawReferenceImage
 - RawReferenceImage.reference_id
 - RawReferenceImage.reference_image
 - RawReferenceImage.reference_type
- RawReferenceImageDict
 - RawReferenceImageDict.reference_id
 - RawReferenceImageDict.reference_image
 - RawReferenceImageDict.reference_type
- ReplayFile
 - ReplayFile.interactions
 - ReplayFile.replay_id
- ReplayFileDialogt
 - ReplayFileDialogt.interactions
 - ReplayFileDialogt.replay_id
- ReplayInteraction
 - ReplayInteraction.request
 - ReplayInteraction.response
- ReplayInteractionDict
 - ReplayInteractionDict.request
 - ReplayInteractionDict.response
- ReplayRequest
 - ReplayRequest.body_segments
 - ReplayRequest.headers
 - ReplayRequest.method
 - ReplayRequest.url
- ReplayRequestDict
 - ReplayRequestDict.body_segments
 - ReplayRequestDict.headers
 - ReplayRequestDict.method
 - ReplayRequestDict.url
- ReplayResponse
 - ReplayResponse.body_segments
 - ReplayResponse.headers
 - ReplayResponse.sdk_response_segments
 - ReplayResponse.status_code
- ReplayResponseDict
 - ReplayResponseDict.body_segments
 - ReplayResponseDict.headers
 - ReplayResponseDict.sdk_response_segments
 - ReplayResponseDict.status_code
- Retrieval
 - Retrieval.disable_attribution

Skip to content l.vertex_ai_search
 l.vertex_rag_store

- RetrievalDict
 - RetrievalDict.disable_attribution
 - RetrievalDict.vertex_ai_search
 - RetrievalDict.vertex_rag_store
- RetrievalMetadata
 - RetrievalMetadata.google_search_dynamic_retrieval_score
- RetrievalMetadataDict
 - RetrievalMetadataDict.google_search_dynamic_retrieval_score
- SafetyAttributes
 - SafetyAttributes.categories
 - SafetyAttributes.scores
- SafetyAttributesDict
 - SafetyAttributesDict.categories
 - SafetyAttributesDict.scores
- SafetyFilterLevel
 - SafetyFilterLevel.BLOCK_LOW_AND ABOVE
 - SafetyFilterLevel.BLOCK_MEDIUM_AND ABOVE
 - SafetyFilterLevel.BLOCK_NONE
 - SafetyFilterLevel.BLOCK_ONLY_HIGH
- SafetyRating
 - SafetyRating.blocked
 - SafetyRating.category
 - SafetyRating.probability
 - SafetyRating.probability_score
 - SafetyRating.severity
 - SafetyRating.severity_score
- SafetyRatingDict
 - SafetyRatingDict.blocked
 - SafetyRatingDict.category
 - SafetyRatingDict.probability
 - SafetyRatingDict.probability_score
 - SafetyRatingDict.severity
 - SafetyRatingDict.severity_score
- SafetySetting
 - SafetySetting.category
 - SafetySetting.method
 - SafetySetting.threshold
- SafetySettingDict
 - SafetySettingDict.category
 - SafetySettingDict.method
 - SafetySettingDict.threshold

[Skip to content](#)

- Schema
 - `Schema.any_of`
 - `Schema.default`
 - `Schema.description`
 - `Schema.enum`
 - `Schema.example`
 - `Schema.format`
 - `Schema.items`
 - `Schema.max_items`
 - `Schema.max_length`
 - `Schema.max_properties`
 - `Schema.maximum`
 - `Schema.min_items`
 - `Schema.min_length`
 - `Schema.min_properties`
 - `Schema.minimum`
 - `Schema.nullable`
 - `Schema.pattern`
 - `Schema.properties`
 - `Schema.property_ordering`
 - `Schema.required`
 - `Schema.title`
 - `Schema.type`
 - SchemaDict
 - `SchemaDict.any_of`
 - `SchemaDict.default`
 - `SchemaDict.description`
 - `SchemaDict.enum`
 - `SchemaDict.example`
 - `SchemaDict.format`
 - `SchemaDict.max_items`
 - `SchemaDict.max_length`
 - `SchemaDict.max_properties`
 - `SchemaDict.maximum`
 - `SchemaDict.min_items`
 - `SchemaDict.min_length`
 - `SchemaDict.min_properties`
 - `SchemaDict.minimum`
 - `SchemaDict.nullable`
 - `SchemaDict.pattern`
 - `SchemaDict.properties`
 - `SchemaDict.property_ordering`
- Skip to content `>t.required`
`>t.title`
▪ `SchemaDict.type`

- SearchEntryPoint
 - SearchEntryPoint.rendered_content
 - SearchEntryPoint.sdk_blob
- SearchEntryPointDict
 - SearchEntryPointDict.rendered_content
 - SearchEntryPointDict.sdk_blob
- Segment
 - Segment.end_index
 - Segment.part_index
 - Segment.start_index
 - Segment.text
- SegmentDict
 - SegmentDict.end_index
 - SegmentDict.part_index
 - SegmentDict.start_index
 - SegmentDict.text
- SpeechConfig
 - SpeechConfig.voice_config
- SpeechConfigDict
 - SpeechConfigDict.voice_config
- State
 - State.ACTIVE
 - State.ERROR
 - State.STATE_UNSPECIFIED
- StyleReferenceConfig
 - StyleReferenceConfig.style_description
- StyleReferenceConfigDict
 - StyleReferenceConfigDict.style_description
- StyleReferenceImage
 - StyleReferenceImage.config
 - StyleReferenceImage.reference_id
 - StyleReferenceImage.reference_image
 - StyleReferenceImage.reference_type
 - StyleReferenceImage.style_image_config
- StyleReferenceImageDict
 - StyleReferenceImageDict.config
 - StyleReferenceImageDict.reference_id
 - StyleReferenceImageDict.reference_image
 - StyleReferenceImageDict.reference_type
- SubjectReferenceConfig
 - SubjectReferenceConfig.subject_description
 - SubjectReferenceConfig.subject_type
- SubjectReferenceConfigDict
 - SubjectReferenceConfigDict.subject_description
 - SubjectReferenceConfigDict.subject_type

Skip to content [referenceConfigDict](#)
[SubjectReferenceConfigDict.subject_description](#)
▪ SubjectReferenceConfigDict.subject_type

- SubjectReferenceImage
 - SubjectReferenceImage.config
 - SubjectReferenceImage.reference_id
 - SubjectReferenceImage.reference_image
 - SubjectReferenceImage.reference_type
 - SubjectReferenceImage.subject_image_config
- SubjectReferenceImageDict
 - SubjectReferenceImageDict.config
 - SubjectReferenceImageDict.reference_id
 - SubjectReferenceImageDict.reference_image
 - SubjectReferenceImageDict.reference_type
- SubjectReferenceType
 - SubjectReferenceType.SUBJECT_TYPE_ANIMAL
 - SubjectReferenceType.SUBJECT_TYPE_DEFAULT
 - SubjectReferenceType.SUBJECT_TYPE_PERSON
 - SubjectReferenceType.SUBJECT_TYPE_PRODUCT
- SupervisedHyperParameters
 - SupervisedHyperParameters.adapter_size
 - SupervisedHyperParameters.epoch_count
 - SupervisedHyperParameters.learning_rate_multiplier
- SupervisedHyperParametersDict
 - SupervisedHyperParametersDict.adapter_size
 - SupervisedHyperParametersDict.epoch_count
 - SupervisedHyperParametersDict.learning_rate_multiplier
- SupervisedTuningDataStats
 - SupervisedTuningDataStats.total_billable_character_count
 - SupervisedTuningDataStats.total_billable_token_count
 - SupervisedTuningDataStats.total_truncated_example_count
 - SupervisedTuningDataStats.total_tuning_character_count
 - SupervisedTuningDataStats.truncated_example_indices
 - SupervisedTuningDataStats.tuning_dataset_example_count
 - SupervisedTuningDataStats.tuning_step_count
 - SupervisedTuningDataStats.user_dataset_examples
 - SupervisedTuningDataStats.user_input_token_distribution
 - SupervisedTuningDataStats.user_message_per_example_distribution
 - SupervisedTuningDataStats.user_output_token_distribution

[Skip to content](#)

- SupervisedTuningDataStatsDict
 - SupervisedTuningDataStatsDict.total_billable_character_count
 - SupervisedTuningDataStatsDict.total_billable_token_count
 - SupervisedTuningDataStatsDict.total_truncated_example_count
 - SupervisedTuningDataStatsDict.total_tuning_character_count
 - SupervisedTuningDataStatsDict.truncated_example_indices
 - SupervisedTuningDataStatsDict.tuning_dataset_example_count
 - SupervisedTuningDataStatsDict.tuning_step_count
 - SupervisedTuningDataStatsDict.user_dataset_examples
 - SupervisedTuningDataStatsDict.user_input_token_distribution
 - SupervisedTuningDataStatsDict.user_message_per_example_distribution
 - SupervisedTuningDataStatsDict.user_output_token_distribution
- SupervisedTuningDatasetDistribution
 - SupervisedTuningDatasetDistribution.billable_sum
 - SupervisedTuningDatasetDistribution.buckets
 - SupervisedTuningDatasetDistribution.max
 - SupervisedTuningDatasetDistribution.mean
 - SupervisedTuningDatasetDistribution.median
 - SupervisedTuningDatasetDistribution.min
 - SupervisedTuningDatasetDistribution.p5
 - SupervisedTuningDatasetDistribution.p95
 - SupervisedTuningDatasetDistribution.sum
- SupervisedTuningDatasetDistributionDatasetBucket
 - SupervisedTuningDatasetDistributionDatasetBucket.count
 - SupervisedTuningDatasetDistributionDatasetBucket.left
 - SupervisedTuningDatasetDistributionDatasetBucket.right
- SupervisedTuningDatasetDistributionDatasetBucketDict
 - SupervisedTuningDatasetDistributionDatasetBucketDict.count
 - SupervisedTuningDatasetDistributionDatasetBucketDict.left
 - SupervisedTuningDatasetDistributionDatasetBucketDict.right
- SupervisedTuningDatasetDistributionDict
 - SupervisedTuningDatasetDistributionDict.billable_sum
 - SupervisedTuningDatasetDistributionDict.buckets
 - SupervisedTuningDatasetDistributionDict.max
 - SupervisedTuningDatasetDistributionDict.mean
 - SupervisedTuningDatasetDistributionDict.median
 - SupervisedTuningDatasetDistributionDict.min
 - SupervisedTuningDatasetDistributionDict.p5
 - SupervisedTuningDatasetDistributionDict.p95
 - SupervisedTuningDatasetDistributionDict.sum
- SupervisedTuningSpec
 - SupervisedTuningSpec.hyper_parameters

Skip to content `>dTuningSpec.training_dataset_uri`
`>dTuningSpec.validation_dataset_uri`

- SupervisedTuningSpecDict
 - `SupervisedTuningSpecDict.hyper_parameters`
 - `SupervisedTuningSpecDict.training_dataset_uri`
 - `SupervisedTuningSpecDict.validation_dataset_uri`
- TestTableFile
 - `TestTableFile.comment`
 - `TestTableFile.parameter_names`
 - `TestTableFile.test_method`
 - `TestTableFile.test_table`
- TestTableFileDialogt
 - `TestTableFileDialogt.comment`
 - `TestTableFileDialogt.parameter_names`
 - `TestTableFileDialogt.test_method`
 - `TestTableFileDialogt.test_table`
- TestTableItem
 - `TestTableItem.exception_if_mldev`
 - `TestTableItem.exception_if_vertex`
 - `TestTableItem.has_union`
 - `TestTableItem.name`
 - `TestTableItem.override_replay_id`
 - `TestTableItem.parameters`
 - `TestTableItem.skip_in_api_mode`
- TestTableItemDict
 - `TestTableItemDict.exception_if_mldev`
 - `TestTableItemDict.exception_if_vertex`
 - `TestTableItemDict.has_union`
 - `TestTableItemDict.name`
 - `TestTableItemDict.override_replay_id`
 - `TestTableItemDict.parameters`
 - `TestTableItemDict.skip_in_api_mode`
- ThinkingConfig
 - `ThinkingConfig.include_thoughts`
- ThinkingConfigDict
 - `ThinkingConfigDict.include_thoughts`
- TokensInfo
 - `TokensInfo.role`
 - `TokensInfo.token_ids`
 - `TokensInfo.tokens`
- TokensInfoDict
 - `TokensInfoDict.role`
 - `TokensInfoDict.token_ids`
 - `TokensInfoDict.tokens`

[Skip to content](#)

- Tool
 - `Tool.code_execution`
 - `Tool.function_declarations`
 - `Tool.google_search`
 - `Tool.google_search_retrieval`
 - `Tool.retrieval`
- ToolCodeExecution
- ToolCodeExecutionDict
- ToolConfig
 - `ToolConfig.function_calling_config`
- ToolConfigDict
 - `ToolConfigDict.function_calling_config`
- ToolDict
 - `ToolDict.code_execution`
 - `ToolDict.function_declarations`
 - `ToolDict.google_search`
 - `ToolDict.google_search_retrieval`
 - `ToolDict.retrieval`
- TunedModel
 - `TunedModel.endpoint`
 - `TunedModel.model`
- TunedModelDict
 - `TunedModelDict.endpoint`
 - `TunedModelDict.model`
- TunedModelInfo
 - `TunedModelInfo.base_model`
 - `TunedModelInfo.create_time`
 - `TunedModelInfo.update_time`
- TunedModelInfoDict
 - `TunedModelInfoDict.base_model`
 - `TunedModelInfoDict.create_time`
 - `TunedModelInfoDict.update_time`
- TuningDataStats
 - `TuningDataStats.distillation_data_stats`
 - `TuningDataStats.supervised_tuning_data_stats`
- TuningDataStatsDict
 - `TuningDataStatsDict.distillation_data_stats`
 - `TuningDataStatsDict.supervised_tuning_data_stats`
- TuningDataset
 - `TuningDataset.examples`
 - `TuningDataset.gcs_uri`
- TuningDatasetDict

Skip to content `:asetDict.examples`
`:asetDict.gcs_uri`

- TuningExample
 - TuningExample.output
 - TuningExample.text_input
- TuningExampleDict
 - TuningExampleDict.output
 - TuningExampleDict.text_input
- TuningJob
 - TuningJob.base_model
 - TuningJob.create_time
 - TuningJob.description
 - TuningJob.distillation_spec
 - TuningJob.encryption_spec
 - TuningJob.end_time
 - TuningJob.error
 - TuningJob.experiment
 - TuningJob.labels
 - TuningJob.name
 - TuningJob.partner_model_tuning_spec
 - TuningJob.pipeline_job
 - TuningJob.start_time
 - TuningJob.state
 - TuningJob.supervised_tuning_spec
 - TuningJob.tuned_model
 - TuningJob.tuned_model_display_name
 - TuningJob.tuning_data_stats
 - TuningJob.update_time
 - TuningJob.hasEnded
 - TuningJob.hasSucceeded

[Skip to content](#)

- TuningJobDict
 - TuningJobDict.base_model
 - TuningJobDict.create_time
 - TuningJobDict.description
 - TuningJobDict.distillation_spec
 - TuningJobDict.encryption_spec
 - TuningJobDict.end_time
 - TuningJobDict.error
 - TuningJobDict.experiment
 - TuningJobDict.labels
 - TuningJobDict.name
 - TuningJobDict.partner_model_tuning_spec
 - TuningJobDict.pipeline_job
 - TuningJobDict.start_time
 - TuningJobDict.state
 - TuningJobDict.supervised_tuning_spec
 - TuningJobDict.tuned_model
 - TuningJobDict.tuned_model_display_name
 - TuningJobDict.tuning_data_stats
 - TuningJobDict.update_time
 - TuningValidationDataset
 - TuningValidationDataset.gcs_uri
 - TuningValidationDatasetDict
 - TuningValidationDatasetDict.gcs_uri
 - Type
 - Type.ARRAY
 - Type.BOOLEAN
 - Type.INTEGER
 - Type.NUMBER
 - Type.OBJECT
 - Type.STRING
 - Type.TYPE_UNSPECIFIED
 - UpdateCachedContentConfig
 - UpdateCachedContentConfig.expire_time
 - UpdateCachedContentConfig.http_options
 - UpdateCachedContentConfig.ttl
 - UpdateCachedContentConfigDict
 - UpdateCachedContentConfigDict.expire_time
 - UpdateCachedContentConfigDict.http_options
 - UpdateCachedContentConfigDict.ttl
 - UpdateModelConfig
 - UpdateModelConfig.description
- Skip to content `telConfig.display_name`
 `telConfig.http_options`

- `UpdateModelConfigDict`
 - `UpdateModelConfigDict.description`
 - `UpdateModelConfigDict.display_name`
 - `UpdateModelConfigDict.http_options`
- `UploadFileConfig`
 - `UploadFileConfig.display_name`
 - `UploadFileConfig.http_options`
 - `UploadFileConfig.mime_type`
 - `UploadFileConfig.name`
- `UploadFileConfigDict`
 - `UploadFileConfigDict.display_name`
 - `UploadFileConfigDict.http_options`
 - `UploadFileConfigDict.mime_type`
 - `UploadFileConfigDict.name`
- `UpscaleImageConfig`
 - `UpscaleImageConfig.http_options`
 - `UpscaleImageConfig.include_rai_reason`
 - `UpscaleImageConfig.output_compression_quality`
 - `UpscaleImageConfig.output_mime_type`
- `UpscaleImageConfigDict`
 - `UpscaleImageConfigDict.http_options`
 - `UpscaleImageConfigDict.include_rai_reason`
 - `UpscaleImageConfigDict.output_compression_quality`
 - `UpscaleImageConfigDict.output_mime_type`
- `UpscaleImageParameters`
 - `UpscaleImageParameters.config`
 - `UpscaleImageParameters.image`
 - `UpscaleImageParameters.model`
 - `UpscaleImageParameters.upscale_factor`
- `UpscaleImageParametersDict`
 - `UpscaleImageParametersDict.config`
 - `UpscaleImageParametersDict.image`
 - `UpscaleImageParametersDict.model`
 - `UpscaleImageParametersDict.upscale_factor`
- `UpscaleImageResponse`
 - `UpscaleImageResponse.generated_images`
- `UpscaleImageResponseDict`
 - `UpscaleImageResponseDict.generated_images`
- `UserContent`
 - `UserContent.parts`
 - `UserContent.role`
- `VertexAISeach`

Skip to content `search.datastore`

`.rchDict`

- `VertexAISeachDict.datastore`

- `VertexRagStore`
 - `VertexRagStore.rag_corpora`
 - `VertexRagStore.rag_resources`
 - `VertexRagStore.similarity_top_k`
 - `VertexRagStore.vector_distance_threshold`
- `VertexRagStoreDict`
 - `VertexRagStoreDict.rag_corpora`
 - `VertexRagStoreDict.rag_resources`
 - `VertexRagStoreDict.similarity_top_k`
 - `VertexRagStoreDict.vector_distance_threshold`
- `VertexRagStoreRagResource`
 - `VertexRagStoreRagResource.rag_corpus`
 - `VertexRagStoreRagResource.rag_file_ids`
- `VertexRagStoreRagResourceDict`
 - `VertexRagStoreRagResourceDict.rag_corpus`
 - `VertexRagStoreRagResourceDict.rag_file_ids`
- `Video`
 - `Video.mime_type`
 - `Video.uri`
 - `Video.video_bytes`
 - `Video.save()`
 - `Video.show()`
- `VideoDict`
 - `VideoDict.mime_type`
 - `VideoDict.uri`
 - `VideoDict.video_bytes`
- `VideoMetadata`
 - `VideoMetadata.end_offset`
 - `VideoMetadata.start_offset`
- `VideoMetadataDict`
 - `VideoMetadataDict.end_offset`
 - `VideoMetadataDict.start_offset`
- `VoiceConfig`
 - `VoiceConfig.prebuilt_voice_config`
- `VoiceConfigDict`
 - `VoiceConfigDict.prebuilt_voice_config`