

Google Gen AI SDK

pypi v0.5.0

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

google-genai is an initial Python client library for interacting with Google's Generative AI APIs. This SDK provides an interface for developers to integrate Google's generative models into their Python applications. This is an early release (version 0.1.0). API is subject to change. Please do not use this SDK in production.

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 (Google AI or Vertex). Feel free to switch the client and run all the examples to see how it behaves under different APIs.

```
# Only run this block for Google AI API
client = genai.Client(api_key='YOUR_API_KEY')
```

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

Types

Skip to content :can be specified as either dictionaries(TypedDict) or pydantic Models. Pydantic available in the `types` module.

Models

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

Generate Content

```
response = client.models.generate_content(  
    model='gemini-2.0-flash-exp', contents='What is your name?'  
)  
print(response.text)
```

System Instructions and Other Configs

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

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-exp',  
    contents=types.Part.from_text('Why is 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,  
    ),  
,
```

Skip to content

Safety Settings

```
response = client.models.generate_content(
    model='gemini-2.0-flash-exp',
    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

Automatic Python function Support

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

```
def get_current_weather(location: str) -> int:
    """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-exp',
    contents="What is the weather like in Boston?",
    config=types.GenerateContentConfig(tools=[get_current_weather],)
)

response.text
```

[Skip to content](#)

```

function = dict(
    name="get_current_weather",
    description="Get the current weather in a given location",
    parameters={
        "type": "OBJECT",
        "properties": {
            "location": {
                "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-exp',
    contents="What is the weather like in Boston?",
    config=types.GenerateContentConfig(tools=[tool]),
)
response.candidates[0].content.parts[0].function_call

```

```

function_call_part = response.candidates[0].content.parts[0]

function_reponse = get_current_weather(**function_call_part.function_call.args)

function_response_part = types.Part.from_function_response(
    name=function_call_part.function_call.name,
    response={'result': function_response}
)

response = client.models.generate_content(
    model='gemini-2.0-flash-exp',
    contents=[
        types.Part.from_text("What is the weather like in Boston?"),
        function_call_part,
        function_response_part,
    ]
)

response

```

[Skip to content](#) [Onse Schema support](#)

Schemas can be 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-exp',
    contents='Give me information of 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-exp',
    contents='Give me information of the United States.',
    config={
        '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

```
print(response.text)
```

Streaming

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

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`

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

print(response.text)
```

Streaming

```
async for response in client.aio.models.generate_content_stream(
    model='gemini-2.0-flash-exp', contents='Tell me a story in 300 words.')
):
    print(response.text)
```

Count Tokens and Compute Tokens

```
response = client.models.count_tokens(
    model='gemini-2.0-flash-exp',
    contents='What is your name?',
)
print(response)
```

Compute Tokens

Compute tokens is not supported by Google AI.

```
response = client.models.compute_tokens(
    model='gemini-2.0-flash-exp',
    contents='What is your name?',
```

Skip to content
)

Async

```
response = await client.aio.models.count_tokens(  
    model='gemini-2.0-flash-exp',  
    contents='What is your name?',  
)  
print(response)
```

Embed Content

```
response = client.models.embed_content(  
    model='text-embedding-004',  
    contents='What is your name?',  
)  
response
```

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

Imagen

Generate Image

Support for generate image in Google AI is behind an allowlist

```
# Generate Image  
response1 = client.models.generate_image(  
    model='imagen-3.0-generate-001',  
    prompt='An umbrella in the foreground, and a rainy night sky in the background',  
    config=types.GenerateImageConfig(  
        negative_prompt= "human",  
        number_of_images= 1,  
        include_rai_reason= True,  
        output_mime_type= "image/jpeg"  
    )  
)  
response1.generated_images[0].image.show()
```

Skip to content ↗

~~generate image is not supported in Google AI.~~

```
# Upscale the generated image from above
response2 = client.models.upscale_image(
    model='imagen-3.0-generate-001',
    image=response1.generated_images[0].image,
    config=types.UpscaleImageConfig(upscale_factor="x2")
)
response2.generated_images[0].image.show()
```

Edit Image

Edit image is not supported in Google AI.

```
# 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-preview-0930',
    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,
        negative_prompt='human',
        include_rai_reason=True,
        output_mime_type='image/jpeg',
    ),
)
response3.generated_images[0].image.show()
```

Files (Only Google AI)

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

[Skip to content](#)

Upload

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

print(file1)
print(file2)
```

Delete

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

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

Caches

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

[Skip to 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',  
    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',)])  
    ],  
    config=types.CreateCachedContentConfig(  
        display_name='test cache',  
        system_instruction='What is the sum of the two pdfs?',  
        ttl='3600s',  
    ),  
)
```

Get

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

Generate Content

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

[Skip to content](#)

Tunings

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

Tune

- Vertex supports tuning from GCS source
- Google AI 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/sf'  
    )  
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)  
        ],  
    )
```

```
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"  
    )  
)  
tuning_job
```

Get Tuning Job

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

[Skip to content](#)

```
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)
```

Use Tuned Model

```
response = client.models.generate_content(
    model=tuning_job.tuned_model.endpoint,
    contents='What is your name?',
)

response.text
```

Get Tuned Model

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

List Tuned Models

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

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

Async

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

Skip to content

```
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])
```

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'))

model
```

Distillation

Only supported on Vertex. Requires allowlist.

```
distillation_job = client.tunings.distill(
    student_model="gemma-2b-1.1-it",
    teacher_model="gemini-1.5-pro-002",
    training_dataset=genai.types.DistillationDataset(
        gcs_uri="gs://cloud-samples-data/ai-platform/generative_ai/gemini-1_5/text/sft"),
    config=genai.types.CreateDistillationJobConfig(
        epoch_count=1,
        pipeline_root_directory=(
            "gs://vertex-sdk-dev-staging-us-central1/tmp/distillation_pipeline_root"
        ),
    ),
)
distillation_job
```

[Skip to content](#)

```
tcompleted_states = set([
    "JOB_STATE_SUCCEEDED",
    "JOB_STATE_FAILED",
    "JOB_STATE_CANCELLED",
    "JOB_STATE_PAUSED"
])
while distillation_job.state not in completed_states:
    print(distillation_job.state)
    distillation_job = client.tunings.get(name=distillation_job.name)
    time.sleep(10)
```

```
distillation_job
```

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

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

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

Batch Prediction

Only supported in Vertex AI.

Skip to content

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

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

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

Async

Skip to content `in await client.aio.batches.list(config={'page_size': 10}):`

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

Delete

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

delete_job
```

[Skip to content](#)

Reference

- [Submodules](#)
- [genai.client module](#)
 - [AsyncClient](#)
 - [AsyncClient.batches](#)
 - [AsyncClient.caches](#)
 - [AsyncClient.chats](#)
 - [AsyncClient.files](#)
 - [AsyncClient.live](#)
 - [AsyncClient.models](#)
 - [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.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\(\)](#)
- [AsyncChats](#)
 - [AsyncChats.create\(\)](#)
- [Chat](#)
 - [Chat.send_message\(\)](#)
- [Chats](#)
 - [Chats.create\(\)](#)

[Skip to content](#)

- [genai.files module](#)

- [AsyncFiles](#)
 - `AsyncFiles.delete()`
 - `AsyncFiles.get()`
 - `AsyncFiles.list()`
 - `AsyncFiles.upload()`
- [Files](#)
 - `Files.delete()`
 - `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_image()`
 - `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_image()`
 - `Models.get()`
 - `Models.list()`
 - `Models.update()`
 - `Models.upscale_image()`

- `genai.tunings module`

- `AsyncTunings`
 - `AsyncTunings.distill()`
 - `AsyncTunings.get()`
 - `AsyncTunings.list()`
 - `AsyncTunings.tune()`
- `Tunings`
 - `Tunings.distill()`
 - `Tunings.get()`
 - `Tunings.list()`
 - `Tunings.tune()`

[Skip to content](#)

- genai.types module

- 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.bigquery_uri
 - BatchJobDestination.format
 - BatchJobDestination.gcs_uri
- BatchJobDestinationDict
 - BatchJobDestinationDict.bigquery_uri
 - BatchJobDestinationDict.format
 - BatchJobDestinationDict.gcs_uri
- 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

Skip to content [dict.update_time](#)

- [BatchJobSource](#)
 - [BatchJobSource.bigrquery_uri](#)
 - [BatchJobSource.format](#)
 - [BatchJobSource.gcs_uri](#)
- [BatchJobSourceDict](#)
 - [BatchJobSourceDict.bigrquery_uri](#)
 - [BatchJobSourceDict.format](#)
 - [BatchJobSourceDict.gcs_uri](#)
- [Blob](#)
 - [Blob.data](#)
 - [Blob.mime_type](#)
- [BlobDict](#)
 - [BlobDict.data](#)
 - [BlobDict.mime_type](#)
- [CachedContent](#)
 - [CachedContent.create_time](#)
 - [CachedContent.display_name](#)
 - [CachedContent.expire_time](#)
 - [CachedContent.model](#)
 - [CachedContent.name](#)
 - [CachedContent.update_time](#)
 - [CachedContent.usage_metadata](#)
- [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](#)

Skip to content [intentUsageMetadataDict.total_token_count](#)
[intentUsageMetadataDict.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](#)
- [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](#)

Skip to content [adata](#)

[..Metadata.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`
- `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`

[Skip to content](#)

- `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`
- `CountTokensConfig`
 - `CountTokensConfig.generation_config`
 - `CountTokensConfig.http_options`
 - `CountTokensConfig.system_instruction`
 - `CountTokensConfig.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.display_name`
 - `CreateCachedContentConfig.expire_time`
 - `CreateCachedContentConfig.http_options`
 - `CreateCachedContentConfig.system_instruction`

Skip to content `>hedContentConfig.tool_config`

- `>hedContentConfig.tools`
 - `CreateCachedContentConfig.ttl`

- `CreateCachedContentConfigDict`
 - `CreateCachedContentConfigDict.display_name`
 - `CreateCachedContentConfigDict.expire_time`
 - `CreateCachedContentConfigDict.http_options`
 - `CreateCachedContentConfigDict.system_instruction`
 - `CreateCachedContentConfigDict.tool_config`
 - `CreateCachedContentConfigDict.tools`
 - `CreateCachedContentConfigDict.ttl`
- `CreateDistillationJobConfig`
 - `CreateDistillationJobConfig.adapter_size`
 - `CreateDistillationJobConfig.epoch_count`
 - `CreateDistillationJobConfig.http_options`
 - `CreateDistillationJobConfig.learning_rate_multiplier`
 - `CreateDistillationJobConfig.pipeline_root_directory`
 - `CreateDistillationJobConfig.tuned_model_display_name`
 - `CreateDistillationJobConfig.validation_dataset`
- `CreateDistillationJobConfigDict`
 - `CreateDistillationJobConfigDict.adapter_size`
 - `CreateDistillationJobConfigDict.epoch_count`
 - `CreateDistillationJobConfigDict.http_options`
 - `CreateDistillationJobConfigDict.learning_rate_multiplier`
 - `CreateDistillationJobConfigDict.pipeline_root_directory`
 - `CreateDistillationJobConfigDict.tuned_model_display_name`
 - `CreateDistillationJobConfigDict.validation_dataset`
- `CreateFileConfig`
 - `CreateFileConfig.http_options`
- `CreateFileConfigDict`
 - `CreateFileConfigDict.http_options`
- `CreateFileResponse`
- `CreateFileResponseDict`
- `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`

[Skip to content](#)

- `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`
- `DatasetDistributionDistributionBucketDict`
 - `DatasetDistributionDistributionBucketDict.count`
 - `DatasetDistributionDistributionBucketDict.left`
 - `DatasetDistributionDistributionBucketDict.right`

[Skip to content](#)

- `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`
- `DeleteCachedContentConfig`
 - `DeleteCachedContentConfig.http_options`
- `DeleteCachedContentConfigDict`
 - `DeleteCachedContentConfigDict.http_options`
- `DeleteCachedContentResponse`
- `DeleteCachedContentResponseDict`
- `DeleteFileConfig`
 - `DeleteFileConfig.http_options`
- `DeleteFileConfigDict`
 - `DeleteFileConfigDict.http_options`
- `DeleteFileResponse`
- `DeleteFileResponseDict`
- `DeleteModelResponse`
- `DeleteModelResponseDict`
- `DeleteResourceJob`
 - `DeleteResourceJob.done`
 - `DeleteResourceJob.error`
 - `DeleteResourceJob.name`
- `DeleteResourceJobDict`
 - `DeleteResourceJobDict.done`
 - `DeleteResourceJobDict.error`
 - `DeleteResourceJobDict.name`

Skip to content [nDataStats](#)

[__init__.py](#) `nDataStats.training_dataset_stats`

- `DistillationDataStatsDict`
 - `DistillationDataStatsDict.training_dataset_stats`
- `DistillationDataset`
 - `DistillationDataset.gcs_uri`
- `DistillationDatasetDict`
 - `DistillationDatasetDict.gcs_uri`
- `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`
- `DistillationValidationDataset`
 - `DistillationValidationDataset.gcs_uri`
- `DistillationValidationDatasetDict`
 - `DistillationValidationDatasetDict.gcs_uri`
- `DynamicRetrievalConfig`
 - `DynamicRetrievalConfig.dynamic_threshold`
 - `DynamicRetrievalConfig.mode`
- `DynamicRetrievalConfigDict`
 - `DynamicRetrievalConfigDict.dynamic_threshold`
 - `DynamicRetrievalConfigDict.mode`

[Skip to content](#)

- `EditImageConfig`
 - `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.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`
 - `EditImageConfigDict.seed`
- `EditImageResponse`
 - `EditImageResponse.generated_images`
- `EditImageResponseDict`
 - `EditImageResponseDict.generated_images`
- `EmbedContentConfig`
 - `EmbedContentConfig.auto_truncate`
 - `EmbedContentConfig.http_options`
 - `EmbedContentConfig.mime_type`
 - `EmbedContentConfig.output_dimensionality`
 - `EmbedContentConfig.task_type`

Skip to content `:entConfig.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`
- `EndpointDict`
 - `EndpointDict.deployed_model_id`
 - `EndpointDict.name`
- `ExecutableCode`
 - `ExecutableCode.code`
 - `ExecutableCode.language`
- `ExecutableCodeDict`
 - `ExecutableCodeDict.code`
 - `ExecutableCodeDict.language`

[Skip to content](#)

- `File`
 - `File.create_time`
 - `File.display_name`
 - `File.error`
 - `File.expiration_time`
 - `File.mime_type`
 - `File.name`
 - `File.sha256_hash`
 - `File.size_bytes`
 - `File.state`
 - `File.update_time`
 - `File.uri`
 - `File.video_metadata`
- `FileData`
 - `FileData.file_uri`
 - `FileData.mime_type`
- `FileDataDict`
 - `FileDataDict.file_uri`
 - `FileDataDict.mime_type`
- `FileDict`
 - `FileDict.create_time`
 - `FileDict.display_name`
 - `FileDict.error`
 - `FileDict.expiration_time`
 - `FileDict.mime_type`
 - `FileDict.name`
 - `FileDict.sha256_hash`
 - `FileDict.size_bytes`
 - `FileDict.state`
 - `FileDict.update_time`
 - `FileDict.uri`
 - `FileDict.video_metadata`
- `FileStatus`
 - `FileStatus.code`
 - `FileStatus.details`
 - `FileStatus.message`
- `FileStatusDict`
 - `FileStatusDict.code`
 - `FileStatusDict.details`
 - `FileStatusDict.message`

[Skip to content](#)

- `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`
- `FunctionDeclaration`
 - `FunctionDeclaration.description`
 - `FunctionDeclaration.name`
 - `FunctionDeclaration.parameters`
 - `FunctionDeclaration.response`
 - `FunctionDeclaration.from_function()`
- `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.automatic_function_calling](#)
 - [GenerateContentConfig.cached_content](#)
 - [GenerateContentConfig.candidate_count](#)
 - [GenerateContentConfig.frequency_penalty](#)
 - [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.tool_config](#)
 - [GenerateContentConfig.tools](#)
 - [GenerateContentConfig.top_k](#)
 - [GenerateContentConfig.top_p](#)

[Skip to content](#)

- `GenerateContentConfigDict`
 - `GenerateContentConfigDict.automatic_function_calling`
 - `GenerateContentConfigDict.cached_content`
 - `GenerateContentConfigDict.candidate_count`
 - `GenerateContentConfigDict.frequency_penalty`
 - `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.tool_config`
 - `GenerateContentConfigDict.tools`
 - `GenerateContentConfigDict.top_k`
 - `GenerateContentConfigDict.top_p`
- `GenerateContentResponse`
 - `GenerateContentResponse.automatic_function_calling_history`
 - `GenerateContentResponse.candidates`
 - `GenerateContentResponse.model_version`
 - `GenerateContentResponse.parsed`
 - `GenerateContentResponse.prompt_feedback`
 - `GenerateContentResponse.usage_metadata`
 - `GenerateContentResponse.text`
- `GenerateContentResponseDict`
 - `GenerateContentResponseDict.candidates`
 - `GenerateContentResponseDict.model_version`
 - `GenerateContentResponseDict.prompt_feedback`
 - `GenerateContentResponseDict.usage_metadata`
- `GenerateContentResponsePromptFeedback`
 - `GenerateContentResponsePromptFeedback.block_reason`
 - `GenerateContentResponsePromptFeedback.block_reason_message`

Skip to content `\ContentResponsePromptFeedback.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`
- `GenerateImageConfig`
 - `GenerateImageConfig.add_watermark`
 - `GenerateImageConfig.aspect_ratio`
 - `GenerateImageConfig.guidance_scale`
 - `GenerateImageConfig.http_options`
 - `GenerateImageConfig.include_rai_reason`
 - `GenerateImageConfig.include_safety_attributes`
 - `GenerateImageConfig.language`
 - `GenerateImageConfig.negative_prompt`
 - `GenerateImageConfig.number_of_images`
 - `GenerateImageConfig.output_compression_quality`
 - `GenerateImageConfig.output_gcs_uri`
 - `GenerateImageConfig.output_mime_type`
 - `GenerateImageConfig.person_generation`
 - `GenerateImageConfig.safety_filter_level`
 - `GenerateImageConfig.seed`

[Skip to content](#)

- `GenerateImageConfigDict`
 - `GenerateImageConfigDict.add_watermark`
 - `GenerateImageConfigDict.aspect_ratio`
 - `GenerateImageConfigDict.guidance_scale`
 - `GenerateImageConfigDict.http_options`
 - `GenerateImageConfigDict.include_rai_reason`
 - `GenerateImageConfigDict.include_safety_attributes`
 - `GenerateImageConfigDict.language`
 - `GenerateImageConfigDict.negative_prompt`
 - `GenerateImageConfigDict.number_of_images`
 - `GenerateImageConfigDict.output_compression_quality`
 - `GenerateImageConfigDict.output_gcs_uri`
 - `GenerateImageConfigDict.output_mime_type`
 - `GenerateImageConfigDict.person_generation`
 - `GenerateImageConfigDict.safety_filter_level`
 - `GenerateImageConfigDict.seed`
- `GenerateImageResponse`
 - `GenerateImageResponse.generated_images`
- `GenerateImageResponseDict`
 - `GenerateImageResponseDict.generated_images`
- `GeneratedImage`
 - `GeneratedImage.image`
 - `GeneratedImage.rai_filtered_reason`
- `GeneratedImageDict`
 - `GeneratedImageDict.image`
 - `GeneratedImageDict.rai_filtered_reason`
- `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`

Skip to content `_onConfig.top_k`

`....._onConfig.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`
- `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`

Skip to content `igDict`

`...configDict.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`
 - `GoogleSearchRetrieval.dynamic_retrieval_config`
- `GoogleSearchRetrievalDict`
 - `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`

Skip to content [unkWeb](#)

[...gChunkWeb.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`
- `GroundingSupport`
 - `GroundingSupport.confidence_scores`
 - `GroundingSupport.grounding_chunk_indices`
 - `GroundingSupport.segment`
- `GroundingSupportDict`
 - `GroundingSupportDict.confidence_scores`
 - `GroundingSupportDict.grounding_chunk_indices`
 - `GroundingSupportDict.segment`
- `Image`
 - `Image.gcs_uri`
 - `Image.image_bytes`
 - `Image.from_file()`
 - `Image.model_post_init()`
 - `Image.save()`
 - `Image.show()`
- `ImageDict`
 - `ImageDict.gcs_uri`
 - `ImageDict.image_bytes`
- `JobError`
 - `JobError.code`
 - `JobError.details`
 - `JobError.message`
- `JobErrorDict`

Skip to content `dict.code`

- `dict.details`
- `JobErrorDict.message`

- ListBatchJobConfig
 - ListBatchJobConfig.filter
 - ListBatchJobConfig.http_options
 - ListBatchJobConfig.page_size
 - ListBatchJobConfig.page_token
 - ListBatchJobConfigDict
 - ListBatchJobConfigDict.filter
 - ListBatchJobConfigDict.http_options
 - ListBatchJobConfigDict.page_size
 - ListBatchJobConfigDict.page_token
 - ListBatchJobResponse
 - ListBatchJobResponse.batch_jobs
 - ListBatchJobResponse.next_page_token
 - ListBatchJobResponseDict
 - ListBatchJobResponseDict.batch_jobs
 - ListBatchJobResponseDict.next_page_token
 - ListCachedContentsConfig
 - ListCachedContentsConfig.page_size
 - ListCachedContentsConfig.page_token
 - ListCachedContentsConfigDict
 - 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
 - ListFilesResponse.files
 - ListFilesResponse.next_page_token
 - ListFilesResponseDict

[Skip to content](#) ↵ ResponseDict.files

```
    sResponseDict.next_page_token
```

- `ListModelsConfig`
 - `ListModelsConfig.filter`
 - `ListModelsConfig.page_size`
 - `ListModelsConfig.page_token`
- `ListModelsConfigDict`
 - `ListModelsConfigDict.filter`
 - `ListModelsConfigDict.page_size`
 - `ListModelsConfigDict.page_token`
- `ListModelsResponse`
 - `ListModelsResponse.models`
 - `ListModelsResponse.next_page_token`
- `ListModelsResponseDict`
 - `ListModelsResponseDict.models`
 - `ListModelsResponseDict.next_page_token`
- `ListTuningJobsConfig`
 - `ListTuningJobsConfig.filter`
 - `ListTuningJobsConfig.page_size`
 - `ListTuningJobsConfig.page_token`
- `ListTuningJobsConfigDict`
 - `ListTuningJobsConfigDict.filter`
 - `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`
 - `LiveClientContentDict.turns`
- `LiveClientMessage`
 - `LiveClientMessage.client_content`
 - `LiveClientMessage.realtime_update`
 - `LiveClientMessage.setup`
 - `LiveClientMessage.tool_response`

[Skip to content](#)

- `LiveClientMessageDict`
 - `LiveClientMessageDict.client_content`
 - `LiveClientMessageDict.realtime_update`
 - `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`
 - `LiveServerContent.model_turn`
 - `LiveServerContent.turn_complete`
- `LiveServerContentDict`
 - `LiveServerContentDict.interrupted`
 - `LiveServerContentDict.model_turn`
 - `LiveServerContentDict.turn_complete`

Skip to content `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`
- `LogprobsResultTopCandidatesDict`
 - `LogprobsResultTopCandidatesDict.candidates`

[Skip to content](#)

- `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`
- `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`
- `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`

Skip to content `::version`

- `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`
- `PrebuiltVoiceConfig`
 - `PrebuiltVoiceConfig.voice_name`
- `PrebuiltVoiceConfigDict`
 - `PrebuiltVoiceConfigDict.voice_name`

[Skip to content](#)

- 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

[Skip to content](#)

- `Retrieval`
 - `Retrieval.disable_attribution`
 - `Retrieval.vertex_ai_search`
 - `Retrieval.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`
- `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](#)

- o 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

[Skip to content](#)

- 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
 - SchemaDict.required
 - SchemaDict.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

Skip to content [gDict](#)

[SpeechConfigDict.voice_config](#)

- `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`
- `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`
- `SupervisedHyperParameters`
 - `SupervisedHyperParameters.adapter_size`
 - `SupervisedHyperParameters.epoch_count`
 - `SupervisedHyperParameters.learning_rate_multiplier`
- `SupervisedHyperParametersDict`
 - `SupervisedHyperParametersDict.adapter_size`
 - `SupervisedHyperParametersDict.epoch_count`
 - `SupervisedHyperParametersDict.learning_rate_multiplier`

[Skip to content](#)

- `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`
- `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`

Skip to content →`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`
 - `SupervisedTuningSpec.training_dataset_uri`
 - `SupervisedTuningSpec.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`
- `TestTableFileDialog`
 - `TestTableFileDialog.comment`
 - `TestTableFileDialog.parameter_names`
 - `TestTableFileDialog.test_method`
 - `TestTableFileDialog.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`

[Skip to content](#)

- `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`
- `TokensInfo`
 - `TokensInfo.role`
 - `TokensInfo.token_ids`
 - `TokensInfo.tokens`
- `TokensInfoDict`
 - `TokensInfoDict.role`
 - `TokensInfoDict.token_ids`
 - `TokensInfoDict.tokens`
- `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`

[Skip to content](#)

- 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
 - TuningDatasetDict.examples
 - TuningDatasetDict.gcs_uri
- TuningExample
 - TuningExample.output
 - TuningExample.text_input
- TuningExampleDict
 - TuningExampleDict.output
 - TuningExampleDict.text_input

[Skip to content](#)

- [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.has_ended](#)
 - [TuningJob.has_succeeded](#)
- [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](#)

Skip to content [Dict.tuning_data_stats](#)

[Dict.update_time](#)

- [TuningJobOrOperation](#)
 - [TuningJobOrOperation.tuning_job](#)
- [TuningJobOrOperationDict](#)
 - [TuningJobOrOperationDict.tuning_job](#)
- [TuningValidationDataset](#)
 - [TuningValidationDataset.gcs_uri](#)
- [TuningValidationDatasetDict](#)
 - [TuningValidationDatasetDict.gcs_uri](#)
- [UpdateCachedContentConfig](#)
 - [UpdateCachedContentConfig.expire_time](#)
 - [UpdateCachedContentConfig.http_options](#)
 - [UpdateCachedContentConfig.ttl](#)
- [UpdateCachedContentConfigDict](#)
 - [UpdateCachedContentConfigDict.expire_time](#)
 - [UpdateCachedContentConfigDict.http_options](#)
 - [UpdateCachedContentConfigDict.ttl](#)
- [UpdateModelConfig](#)
 - [UpdateModelConfig.description](#)
 - [UpdateModelConfig.display_name](#)
- [UpdateModelConfigDict](#)
 - [UpdateModelConfigDict.description](#)
 - [UpdateModelConfigDict.display_name](#)
- [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](#)
 - [UpscaleImageConfig.upscale_factor](#)

[Skip to content](#)

- `UpscaleImageConfigDict`
 - `UpscaleImageConfigDict.http_options`
 - `UpscaleImageConfigDict.include_rai_reason`
 - `UpscaleImageConfigDict.output_compression_quality`
 - `UpscaleImageConfigDict.output_mime_type`
 - `UpscaleImageConfigDict.upscale_factor`
- `UpscaleImageParameters`
 - `UpscaleImageParameters.config`
 - `UpscaleImageParameters.image`
 - `UpscaleImageParameters.model`
- `UpscaleImageParametersDict`
 - `UpscaleImageParametersDict.config`
 - `UpscaleImageParametersDict.image`
 - `UpscaleImageParametersDict.model`
- `UpscaleImageResponse`
 - `UpscaleImageResponse.generated_images`
- `UpscaleImageResponseDict`
 - `UpscaleImageResponseDict.generated_images`
- `VertexAISeach`
 - `VertexAISeach.datastore`
- `VertexAISeachDict`
 - `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`
- `VideoMetadata`
 - `VideoMetadata.end_offset`

Skip to content `\data.start_offset`

- `VideoMetadataDict`
 - `VideoMetadataDict.end_offset`
 - `VideoMetadataDict.start_offset`
- `VoiceConfig`
 - `VoiceConfig.prebuilt_voice_config`
- `VoiceConfigDict`
 - `VoiceConfigDict.prebuilt_voice_config`

Copyright © 2024, Google
Made with Sphinx and @pradyunsg's Furo