Gemini 2.5 Flash Image (aka Nano Banana) is now available in the Gemini API!

Learn more (https://ai.google.dev/gemini-api/docs/image-generation#image_generation_text-to-image)

Generating content

The Gemini API supports content generation with images, audio, code, tools, and more. For details on each of these features, read on and check out the task-focused sample code, or read the comprehensive guides.

- Text generation (https://ai.google.dev/gemini-api/docs/text-generation)
- Vision (https://ai.google.dev/gemini-api/docs/vision)
- Audio (https://ai.google.dev/gemini-api/docs/audio)
- Embeddings (https://ai.google.dev/gemini-api/docs/embeddings)
- Long context (https://ai.google.dev/gemini-api/docs/long-context)
- Code execution (https://ai.google.dev/gemini-api/docs/code-execution)
- JSON Mode (https://ai.google.dev/gemini-api/docs/json-mode)
- Function calling (https://ai.google.dev/gemini-api/docs/function-calling)
- System instructions (https://ai.google.dev/gemini-api/docs/system-instructions)

Method: models.generateContent

Generates a model response given an input `GenerateContentRequest`. Refer to the text generation guide (https://ai.google.dev/gemini-api/docs/text-generation) for detailed usage information. Input capabilities differ between models, including tuned models. Refer to the model guide (https://ai.google.dev/gemini-api/docs/models/gemini) and tuning guide (https://ai.google.dev/gemini-api/docs/model-tuning) for details.

Endpoint

```
POST https://generativelanguage.googleapis.com/v1beta/{model=models/*}:generateContent
```

Path parameters

`model` `string`

Required. The name of the `Model` to use for generating the completion.
Format: `models/{model}`. It takes the form `models/{model}`.

Request body

The request body contains data with the following structure:

Fields

`contents[]` `object` (`Content` (/api/caching#Content))

Required. The content of the current conversation with the model.
For single-turn queries, this is a single instance. For multi-turn queries like chat (https://ai.google.dev/gemini-api/docs/text-generation#chat), this is a repeated field that contains the conversation history and the latest request.

`tools[]` `object` (`Tool` (/api/caching#Tool))

Optional. A list of `Tools` the `Model` may use to generate the next response.
A `Tool` is a piece of code that enables the system to interact with external systems to perform an action, or set of actions, outside of knowledge and scope of the `Model`. Supported `Tools` are `Function` and `codeExecution`. Refer to the Function calling (https://ai.google.dev/gemini-api/docs/function-calling) and the Code execution (https://ai.google.dev/gemini-api/docs/code-execution) guides to learn more.

`toolConfig` `object` (`ToolConfig` (/api/caching#ToolConfig))

Optional. Tool configuration for any `Tool` specified in the request. Refer to the Function calling guide (https://ai.google.dev/gemini-api/docs/function-calling#function_calling_mode) for a usage example.

`safetySettings[]` `object` (`SafetySetting` (/api/generate-content#v1beta.SafetySetting))

Optional. A list of unique `SafetySetting` instances for blocking unsafe content.
This will be enforced on the `GenerateContentRequest.contents` and `GenerateContentResponse.candidates`. There should not be more than one setting for each `SafetyCategory` type. The API will block any contents and responses that fail to meet the thresholds set by these settings. This list overrides the default settings for each `SafetyCategory` specified in the safetySettings. If there is no `SafetySetting` for a given `SafetyCategory` provided in the list, the API will use the default safety setting for that category. Harm categories HARM_CATEGORY_HATE_SPEECH, HARM_CATEGORY_SEXUALLY_EXPLICIT, HARM_CATEGORY_DANGEROUS_CONTENT, HARM_CATEGORY_HARASSMENT, HARM_CATEGORY_CIVIC_INTEGRITY are supported. Refer to the guide (https://ai.google.dev/gemini-api/docs/safety-settings) for detailed information on available safety settings. Also refer to the Safety guidance (https://ai.google.dev/gemini-api/docs/safety-guidance) to learn how to incorporate safety considerations in your AI applications.

systemInstruction object (Content (/api/caching#Content))

Optional. Developer set system instruction(s) (https://ai.google.dev/gemini-api/docs/system-instructions). Currently, text only.

generationConfig object (GenerationConfig (/api/generate-content#v1beta.GenerationConfig))

Optional. Configuration options for model generation and outputs.

cachedContent string

Optional. The name of the content cached (https://ai.google.dev/gemini-api/docs/caching) to use as context to serve the prediction. Format: `cachedContents/{cachedContent}`

Example request

Text (#text) Image (#image) Audio (#audio) Video (#video) PDF (#pdf) More

Python (#python) Node.js (#node.js) Go (#go) Shell (#shell) Java (#java)

```
from google import genai

client = genai.Client()
response = client.models.generate_content(
    model="gemini-2.0-flash", contents="Write a story about a magic back
)
print(response.text)
```

/9f5adb78a77820ef2d4f2a040d698481803e8214/python/text_generation.py#L26-L32)

Response body

If successful, the response body contains an instance of **GenerateContentResponse** (/api/generate-content#v1beta.GenerateContentResponse).

Method: models.streamGenerateContent

Generates a streamed response (https://ai.google.dev/gemini-api/docs/text-generation?lang=python#generate-a-text-stream) from the model given an input `GenerateContentRequest`.

Endpoint

```
POST https://generativelanguage.googleapis.com/v1beta/{model=models/*}:streamGenerateContent
```

Path parameters

`model` `string`

Required. The name of the `Model` to use for generating the completion.
Format: `models/{model}`. It takes the form `models/{model}`.

Request body

The request body contains data with the following structure:

Fields

`contents[]` `object` (`Content` (/api/caching#Content))

Required. The content of the current conversation with the model.
For single-turn queries, this is a single instance. For multi-turn queries like chat (https://ai.google.dev/gemini-api/docs/text-generation#chat), this is a repeated field that contains the conversation history and the latest request.

`tools[]` `object` (`Tool` (/api/caching#Tool))

Optional. A list of `Tools` the `Model` may use to generate the next response.
A `Tool` is a piece of code that enables the system to interact with external systems to perform an action, or set of actions, outside of knowledge and scope of the `Model`. Supported `Tools` are `Function` and `codeExecution`. Refer to the Function calling (https://ai.google.dev/gemini-api/docs/function-calling) and the Code execution (https://ai.google.dev/gemini-api/docs/code-execution) guides to learn more.

`toolConfig` `object` (`ToolConfig` (/api/caching#ToolConfig))

Optional. Tool configuration for any `Tool` specified in the request. Refer to the Function calling guide (https://ai.google.dev/gemini-api/docs/function-calling#function_calling_mode) for a usage example.

`safetySettings[]` `object` (`SafetySetting` (/api/generate-content#v1beta.SafetySetting))

Optional. A list of unique `SafetySetting` instances for blocking unsafe content.

This will be enforced on the `GenerateContentRequest.contents` and `GenerateContentResponse.candidates`. There should not be more than one setting for each `SafetyCategory` type. The API will block any contents and responses that fail to meet the thresholds set by these settings. This list overrides the default settings for each `SafetyCategory` specified in the safetySettings. If there is no `SafetySetting` for a given `SafetyCategory` provided in the list, the API will use the default safety setting for that category. Harm categories HARM_CATEGORY_HATE_SPEECH, HARM_CATEGORY_SEXUALLY_EXPLICIT, HARM_CATEGORY_DANGEROUS_CONTENT, HARM_CATEGORY_HARASSMENT, HARM_CATEGORY_CIVIC_INTEGRITY are supported. Refer to the guide (https://ai.google.dev/gemini-api/docs/safety-settings) for detailed information on available safety settings. Also refer to the Safety guidance (https://ai.google.dev/gemini-api/docs/safety-guidance) to learn how to incorporate safety considerations in your AI applications.

`systemInstruction` `object (Content (/api/caching#Content))`

Optional. Developer set system instruction(s) (https://ai.google.dev/gemini-api/docs/system-instructions). Currently, text only.

`generationConfig` `object (GenerationConfig (/api/generate-content#v1beta.GenerationConfig))`

Optional. Configuration options for model generation and outputs.

`cachedContent` `string`

Optional. The name of the content cached (https://ai.google.dev/gemini-api/docs/caching) to use as context to serve the prediction. Format: `cachedContents/{cachedContent}`

Example request

Text Image (#image) Audio (#audio) Video (#video) PDF (#pdf) Chat (#chat)
(#text)

Python Node.js (#node.js) Go (#go) Shell (#shell) Java (#java)
(#python)

```
from google import genai

client = genai.Client()
response = client.models.generate_content_stream(
    model="gemini-2.0-flash", contents="Write a story about a magic back
)
for chunk in response:
    print(chunk.text)
    print("_" * 80)
```

/9f5adb78a77820ef2d4f2a040d698481803e8214/python/text_generation.py#L37-L45)

Response body

If successful, the response body contains a stream of `GenerateContentResponse` (/api/generate-content#v1beta.GenerateContentResponse) instances.

GenerateContentResponse

Response from the model supporting multiple candidate responses.

Safety ratings and content filtering are reported for both prompt in `GenerateContentResponse.prompt_feedback` and for each candidate in `finishReason` and in `safetyRatings`. The API: - Returns either all requested candidates or none of them - Returns no candidates at all only if there was something wrong with the prompt (check `promptFeedback`) - Reports feedback on each candidate in `finishReason` and `safetyRatings`.

Fields

`candidates[]` `object` (`Candidate` (/api/generate-content#v1beta.Candidate))

Candidate responses from the model.

`promptFeedback` `object` (`PromptFeedback` (/api/generate-content#PromptFeedback))

Returns the prompt's feedback related to the content filters.

`usageMetadata` `object` (`UsageMetadata` (/api/generate-content#UsageMetadata))

Output only. Metadata on the generation requests' token usage.

`modelVersion` `string`

Output only. The model version used to generate the response.

`responseId` `string`

Output only. responseId is used to identify each response.

JSON representation

```
{
  "candidates": [
    {
```

JSON representation

```
      object (Candidate (/api/generate-content#v1beta.Candidate))
    }
  ],
  "promptFeedback": {
    object (PromptFeedback (/api/generate-content#PromptFeedback))
  },
  "usageMetadata": {
    object (UsageMetadata (/api/generate-content#UsageMetadata))
  },
  "modelVersion": string,
  "responseId": string
}
```

PromptFeedback

A set of the feedback metadata the prompt specified in `GenerateContentRequest.content`.

Fields

`blockReason` `enum (BlockReason (/api/generate-content#BlockReason))`

Optional. If set, the prompt was blocked and no candidates are returned. Rephrase the prompt.

`safetyRatings[]` `object (SafetyRating (/api/generate-content#v1beta.SafetyRating))`

Ratings for safety of the prompt. There is at most one rating per category.

JSON representation

```
{
  "blockReason": enum (BlockReason (/api/generate-content#BlockReason)),
  "safetyRatings": [
    {
      object (SafetyRating (/api/generate-content#v1beta.SafetyRating))
    }
  ]
}
```

BlockReason

Specifies the reason why the prompt was blocked.

Enums	
`BLOCK_REASON_UNSPECIFIED`	Default value. This value is unused.
`SAFETY`	Prompt was blocked due to safety reasons. Inspect `safetyRatings` to understand which safety category blocked it.
`OTHER`	Prompt was blocked due to unknown reasons.
`BLOCKLIST`	Prompt was blocked due to the terms which are included from the terminology blocklist.
`PROHIBITED_CONTENT`	Prompt was blocked due to prohibited content.
`IMAGE_SAFETY`	Candidates blocked due to unsafe image generation content.

UsageMetadata

Metadata on the generation request's token usage.

Fields

`promptTokenCount` `integer`

Number of tokens in the prompt. When `cachedContent` is set, this is still the total effective prompt size meaning this includes the number of tokens in the cached content.

`cachedContentTokenCount` `integer`

Number of tokens in the cached part of the prompt (the cached content)

`candidatesTokenCount` `integer`

Total number of tokens across all the generated response candidates.

`toolUsePromptTokenCount` `integer`

Output only. Number of tokens present in tool-use prompt(s).

`thoughtsTokenCount` `integer`

Output only. Number of tokens of thoughts for thinking models.

`totalTokenCount` `integer`

Total token count for the generation request (prompt + response candidates).

`promptTokensDetails[]`

`object` (`ModalityTokenCount` (/api/generate-content#v1beta.ModalityTokenCount))

Output only. List of modalities that were processed in the request input.

`cacheTokensDetails[]`

`object` (`ModalityTokenCount` (/api/generate-content#v1beta.ModalityTokenCount))

Output only. List of modalities of the cached content in the request input.

`candidatesTokensDetails[]`

`object` (`ModalityTokenCount` (/api/generate-content#v1beta.ModalityTokenCount))

Output only. List of modalities that were returned in the response.

`toolUsePromptTokensDetails[]`

`object` (`ModalityTokenCount` (/api/generate-content#v1beta.ModalityTokenCount))

Output only. List of modalities that were processed for tool-use request inputs.

JSON representation

```
{
  "promptTokenCount": integer,
  "cachedContentTokenCount": integer,
  "candidatesTokenCount": integer,
  "toolUsePromptTokenCount": integer,
  "thoughtsTokenCount": integer,
  "totalTokenCount": integer,
  "promptTokensDetails": [
    {
      object (ModalityTokenCount (/api/generate-content#v1beta.ModalityTokenCount))
    }
  ],
  "cacheTokensDetails": [
    {
      object (ModalityTokenCount (/api/generate-content#v1beta.ModalityTokenCount))
    }
  ],
  "candidatesTokensDetails": [
    {
      object (ModalityTokenCount (/api/generate-content#v1beta.ModalityTokenCount))
    }
```

JSON representation

```
  ],
  "toolUsePromptTokensDetails": [
    {
      object (ModalityTokenCount (/api/generate-content#v1beta.ModalityTokenCount))
    }
  ]
}
```

Candidate

A response candidate generated from the model.

Fields

content `object (Content (/api/caching#Content))`

Output only. Generated content returned from the model.

finishReason `enum (FinishReason (/api/generate-content#FinishReason))`

Optional. Output only. The reason why the model stopped generating tokens.
If empty, the model has not stopped generating tokens.

safetyRatings[] `object (SafetyRating (/api/generate-content#v1beta.SafetyRating))`

List of ratings for the safety of a response candidate.
There is at most one rating per category.

citationMetadata `object (CitationMetadata (/api/generate-content#v1beta.CitationMetadata))`

Output only. Citation information for model-generated candidate.
This field may be populated with recitation information for any text included in the `content`. These are passages that are "recited" from copyrighted material in the foundational LLM's training data.

tokenCount `integer`

Output only. Token count for this candidate.

groundingAttributions[]

`object (GroundingAttribution (/api/generate-content#GroundingAttribution))`

Output only. Attribution information for sources that contributed to a grounded answer.
This field is populated for `GenerateAnswer` calls.

groundingMetadata `object (GroundingMetadata (/api/generate-content#GroundingMetadata))`

Output only. Grounding metadata for the candidate.
This field is populated for `GenerateContent` calls.

avgLogprobs `number`

Output only. Average log probability score of the candidate.

logprobsResult `object (LogprobsResult (/api/generate-content#LogprobsResult))`

Output only. Log-likelihood scores for the response tokens and top tokens

urlContextMetadata `object (UrlContextMetadata (/api/generate-content#UrlContextMetadata))`

Output only. Metadata related to url context retrieval tool.

index `integer`

Output only. Index of the candidate in the list of response candidates.

JSON representation

```
{
  "content": {
    object (Content (/api/caching#Content))
  },
  "finishReason": enum (FinishReason (/api/generate-content#FinishReason)),
  "safetyRatings": [
    {
      object (SafetyRating (/api/generate-content#v1beta.SafetyRating))
    }
  ],
  "citationMetadata": {
    object (CitationMetadata (/api/generate-content#v1beta.CitationMetadata))
  },
  "tokenCount": integer,
  "groundingAttributions": [
    {
      object (GroundingAttribution (/api/generate-content#GroundingAttribution))
    }
  ],
  "groundingMetadata": {
    object (GroundingMetadata (/api/generate-content#GroundingMetadata))
  },
  "avgLogprobs": number,
  "logprobsResult": {
    object (LogprobsResult (/api/generate-content#LogprobsResult))
```

JSON representation

```
  },
  "urlContextMetadata": {
    object (UrlContextMetadata (/api/generate-content#UrlContextMetadata))
  },
```

FinishReason

Defines the reason why the model stopped generating tokens.

Enums	
`FINISH_REASON_UNSPECIFIED`	Default value. This value is unused.
`STOP`	Natural stop point of the model or provided stop sequence.
`MAX_TOKENS`	The maximum number of tokens as specified in the request was reached.
`SAFETY`	The response candidate content was flagged for safety reasons.
`RECITATION`	The response candidate content was flagged for recitation reasons.
`LANGUAGE`	The response candidate content was flagged for using an unsupported language.
`OTHER`	Unknown reason.
`BLOCKLIST`	Token generation stopped because the content contains forbidden terms.
`PROHIBITED_CONTENT`	Token generation stopped for potentially containing prohibited content.
`SPII`	Token generation stopped because the content potentially contains Sensitive Personally Identifiable Information (SPII).
`MALFORMED_FUNCTION_CALL`	The function call generated by the model is invalid.
`IMAGE_SAFETY`	Token generation stopped because generated images contain safety violations.
`UNEXPECTED_TOOL_CALL`	Model generated a tool call but no tools were enabled in the request.
`TOO_MANY_TOOL_CALLS`	Model called too many tools consecutively, thus the system exited execution.

GroundingAttribution

Attribution for a source that contributed to an answer.

Fields

sourceId `object (AttributionSourceId (/api/generate-content#AttributionSourceId))`

Output only. Identifier for the source contributing to this attribution.

content `object (Content (/api/caching#Content))`

Grounding source content that makes up this attribution.

JSON representation

```
{
  "sourceId": {
    object (AttributionSourceId (/api/generate-content#AttributionSourceId))
  },
  "content": {
    object (Content (/api/caching#Content))
  }
}
```

AttributionSourceId

Identifier for the source contributing to this attribution.

Fields

source `Union type`

source can be only one of the following:

> **groundingPassage** `object (GroundingPassageId (/api/generate-content#GroundingPassageId))`
>
> Identifier for an inline passage.

> **semanticRetrieverChunk**

> `object (SemanticRetrieverChunk` (/api/generate-content#SemanticRetrieverChunk)`)`
>
> Identifier for a `Chunk` fetched via Semantic Retriever.

JSON representation

```
{

  // source
  "groundingPassage": {
    object (GroundingPassageId (/api/generate-content#GroundingPassageId))
  },
  "semanticRetrieverChunk": {
    object (SemanticRetrieverChunk (/api/generate-content#SemanticRetrieverChunk))
  }
  // Union type
}
```

GroundingPassageId

Identifier for a part within a `GroundingPassage`.

Fields

`passageId` `string`

Output only. ID of the passage matching the `GenerateAnswerRequest`'s `GroundingPassage.id`.

`partIndex` `integer`

Output only. Index of the part within the `GenerateAnswerRequest`'s `GroundingPassage.content`.

JSON representation

```
{
  "passageId": string,
  "partIndex": integer
}
```

SemanticRetrieverChunk

Identifier for a `Chunk` retrieved via Semantic Retriever specified in the `GenerateAnswerRequest` using `SemanticRetrieverConfig`.

Fields

`source` `string`

Output only. Name of the source matching the request's `SemanticRetrieverConfig.source`. Example: `corpora/123` or `corpora/123/documents/abc`

`chunk` `string`

Output only. Name of the `Chunk` containing the attributed text. Example: `corpora/123/documents/abc/chunks/xyz`

JSON representation

```
{
  "source": string,
  "chunk": string
}
```

GroundingMetadata

Metadata returned to client when grounding is enabled.

Fields

`groundingChunks[]` `object (GroundingChunk (/api/generate-content#GroundingChunk))`

List of supporting references retrieved from specified grounding source.

`groundingSupports[]` `object (GroundingSupport (/api/generate-content#GroundingSupport))`

List of grounding support.

`webSearchQueries[]` `string`

Web search queries for the following-up web search.

`searchEntryPoint` `object (SearchEntryPoint (/api/generate-content#SearchEntryPoint))`

Optional. Google search entry for the following-up web searches.

retrievalMetadata `object (RetrievalMetadata (/api/generate-content#RetrievalMetadata))`

Metadata related to retrieval in the grounding flow.

JSON representation

```
{
  "groundingChunks": [
    {
      object (GroundingChunk (/api/generate-content#GroundingChunk))
    }
  ],
  "groundingSupports": [
    {
      object (GroundingSupport (/api/generate-content#GroundingSupport))
    }
  ],
  "webSearchQueries": [
    string
  ],
  "searchEntryPoint": {
    object (SearchEntryPoint (/api/generate-content#SearchEntryPoint))
  },
  "retrievalMetadata": {
    object (RetrievalMetadata (/api/generate-content#RetrievalMetadata))
  }
}
```

SearchEntryPoint

Google search entry point.

Fields

renderedContent `string`

Optional. Web content snippet that can be embedded in a web page or an app webview.

sdkBlob `string (bytes (https://developers.google.com/discovery/v1/type-format) format)`

Optional. Base64 encoded JSON representing array of <search term, search url> tuple.
A base64-encoded string.

JSON representation

```
{
  "renderedContent": string,
  "sdkBlob": string
}
```

GroundingChunk

Grounding chunk.

Fields

Chunk type. **chunk_type** can be only one of the following:	
chunk_type Union type	**web** object (Web (/api/generate-content#Web)) Grounding chunk from the web.

JSON representation

```
{

  // chunk_type
  "web": {
    object (Web (/api/generate-content#Web))
  }
  // Union type
}
```

Web

Chunk from the web.

Fields

uri string

URI reference of the chunk.

`title` `string`

Title of the chunk.

JSON representation

```
{
  "uri": string,
  "title": string
}
```

GroundingSupport

Grounding support.

Fields

`groundingChunkIndices[]` `integer`

A list of indices (into 'grounding_chunk') specifying the citations associated with the claim. For instance [1,3,4] means that grounding_chunk[1], grounding_chunk[3], grounding_chunk[4] are the retrieved content attributed to the claim.

`confidenceScores[]` `number`

Confidence score of the support references. Ranges from 0 to 1. 1 is the most confident. This list must have the same size as the groundingChunkIndices.

`segment` `object (Segment (/api/generate-content#Segment))`

Segment of the content this support belongs to.

JSON representation

```
{
  "groundingChunkIndices": [
    integer
  ],
  "confidenceScores": [
    number
  ],
  "segment": {
```

JSON representation

```
    object (Segment (/api/generate-content#Segment))
  }
}
```

Segment

Segment of the content.

Fields

partIndex integer

Output only. The index of a Part object within its parent Content object.

startIndex integer

Output only. Start index in the given Part, measured in bytes. Offset from the start of the Part, inclusive, starting at zero.

endIndex integer

Output only. End index in the given Part, measured in bytes. Offset from the start of the Part, exclusive, starting at zero.

text string

Output only. The text corresponding to the segment from the response.

JSON representation

```
{
  "partIndex": integer,
  "startIndex": integer,
  "endIndex": integer,
  "text": string
}
```

RetrievalMetadata

Metadata related to retrieval in the grounding flow.

Fields

`googleSearchDynamicRetrievalScore` `number`

Optional. Score indicating how likely information from google search could help answer the prompt. The score is in the range [0, 1], where 0 is the least likely and 1 is the most likely. This score is only populated when google search grounding and dynamic retrieval is enabled. It will be compared to the threshold to determine whether to trigger google search.

JSON representation

```
{
  "googleSearchDynamicRetrievalScore": number
}
```

LogprobsResult

Logprobs Result

Fields

`topCandidates[]` `object (TopCandidates (/api/generate-content#TopCandidates))`

Length = total number of decoding steps.

`chosenCandidates[]` `object (Candidate (/api/generate-content#Candidate))`

Length = total number of decoding steps. The chosen candidates may or may not be in topCandidates.

JSON representation

```
{
  "topCandidates": [
    {
      object (TopCandidates (/api/generate-content#TopCandidates))
    }
  ],
  "chosenCandidates": [
    {
      object (Candidate (/api/generate-content#Candidate))
    }
```

JSON representation

```
  ]
}
```

TopCandidates

Candidates with top log probabilities at each decoding step.

Fields

`candidates[]` `object (Candidate (/api/generate-content#Candidate))`

Sorted by log probability in descending order.

JSON representation

```
{
  "candidates": [
    {
      object (Candidate (/api/generate-content#Candidate))
    }
  ]
}
```

Candidate

Candidate for the logprobs token and score.

Fields

`token` `string`

The candidate's token string value.

`tokenId` `integer`

The candidate's token id value.

`logProbability` `number`

The candidate's log probability.

JSON representation

```
{
  "token": string,
  "tokenId": integer,
  "logProbability": number
}
```

UrlContextMetadata

Metadata related to url context retrieval tool.

Fields

`urlMetadata[]` `object (UrlMetadata (/api/generate-content#UrlMetadata))`

List of url context.

JSON representation

```
{
  "urlMetadata": [
    {
      object (UrlMetadata (/api/generate-content#UrlMetadata))
    }
  ]
}
```

UrlMetadata

Context of the a single url retrieval.

Fields

`retrievedUrl` `string`

Retrieved url by the tool.

urlRetrievalStatus `enum (UrlRetrievalStatus (/api/generate-content#UrlRetrievalStatus))`

Status of the url retrieval.

JSON representation

```
{
  "retrievedUrl": string,
  "urlRetrievalStatus": enum (UrlRetrievalStatus (/api/generate-content#UrlRetrievalS
}
```

UrlRetrievalStatus

Status of the url retrieval.

Enums

URL_RETRIEVAL_STATUS_ UNSPECIFIED	Default value. This value is unused.
URL_RETRIEVAL_STATUS_ SUCCESS	Url retrieval is successful.
URL_RETRIEVAL_STATUS_ ERROR	Url retrieval is failed due to error.
URL_RETRIEVAL_STATUS_ PAYWALL	Url retrieval is failed because the content is behind paywall.
URL_RETRIEVAL_STATUS_ UNSAFE	Url retrieval is failed because the content is unsafe.

CitationMetadata

A collection of source attributions for a piece of content.

Fields

citationSources[] `object (CitationSource (/api/generate-content#CitationSource))`

Citations to sources for a specific response.

JSON representation

```
{
  "citationSources": [
    {
      object (CitationSource (/api/generate-content#CitationSource))
    }
  ]
}
```

CitationSource

A citation to a source for a portion of a specific response.

Fields

`startIndex` `integer`

Optional. Start of segment of the response that is attributed to this source.
Index indicates the start of the segment, measured in bytes.

`endIndex` `integer`

Optional. End of the attributed segment, exclusive.

`uri` `string`

Optional. URI that is attributed as a source for a portion of the text.

`license` `string`

Optional. License for the GitHub project that is attributed as a source for segment.
License info is required for code citations.

JSON representation

```
{
  "startIndex": integer,
  "endIndex": integer,
  "uri": string,
  "license": string
}
```

GenerationConfig

Configuration options for model generation and outputs. Not all parameters are configurable for every model.

Fields

`stopSequences[]` `string`

Optional. The set of character sequences (up to 5) that will stop output generation. If specified, the API will stop at the first appearance of a `stop_sequence`. The stop sequence will not be included as part of the response.

`responseMimeType` `string`

Optional. MIME type of the generated candidate text. Supported MIME types are: `text/plain`: (default) Text output. `application/json`: JSON response in the response candidates. `text/x.enum`: ENUM as a string response in the response candidates. Refer to the docs (https://ai.google.dev/gemini-api/docs/prompting_with_media#plain_text_formats) for a list of all supported text MIME types.

`responseSchema` `object` (Schema (/api/caching#Schema))

Optional. Output schema of the generated candidate text. Schemas must be a subset of the OpenAPI schema (https://spec.openapis.org/oas/v3.0.3#schema) and can be objects, primitives or arrays.
If set, a compatible `responseMimeType` must also be set. Compatible MIME types: `application/json`: Schema for JSON response. Refer to the JSON text generation guide (https://ai.google.dev/gemini-api/docs/json-mode) for more details.

`responseJsonSchema`

`value` (Value (https://protobuf.dev/reference/protobuf/google.protobuf/#value) `format`)

Optional. Output schema of the generated response. This is an alternative to `responseSchema` that accepts JSON Schema (https://json-schema.org/).
If set, `responseSchema` must be omitted, but `responseMimeType` is required.
While the full JSON Schema may be sent, not all features are supported. Specifically, only the following properties are supported:

- `$id`
- `$defs`
- `$ref`
- `$anchor`
- `type`

- `format`
- `title`
- `description`
- `enum` (for strings and numbers)
- `items`
- `prefixItems`
- `minItems`
- `maxItems`
- `minimum`
- `maximum`
- `anyOf`
- `oneOf` (interpreted the same as `anyOf`)
- `properties`
- `additionalProperties`
- `required`

The non-standard `propertyOrdering` property may also be set.
Cyclic references are unrolled to a limited degree and, as such, may only be used within non-required properties. (Nullable properties are not sufficient.) If `$ref` is set on a sub-schema, no other properties, except for than those starting as a `$`, may be set.

`responseModalities[]` `enum` (`Modality` (/api/generate-content#Modality))

Optional. The requested modalities of the response. Represents the set of modalities that the model can return, and should be expected in the response. This is an exact match to the modalities of the response.
A model may have multiple combinations of supported modalities. If the requested modalities do not match any of the supported combinations, an error will be returned.
An empty list is equivalent to requesting only text.

`candidateCount` `integer`

Optional. Number of generated responses to return. If unset, this will default to 1. Please note that this doesn't work for previous generation models (Gemini 1.0 family)

`maxOutputTokens` `integer`

Optional. The maximum number of tokens to include in a response candidate.

Note: The default value varies by model, see the `Model.output_token_limit` attribute of the `Model` returned from the `getModel` function.

`temperature` `number`

Optional. Controls the randomness of the output.
Note: The default value varies by model, see the `Model.temperature` attribute of the `Model` returned from the `getModel` function.
Values can range from [0.0, 2.0].

`topP` `number`

Optional. The maximum cumulative probability of tokens to consider when sampling.
The model uses combined Top-k and Top-p (nucleus) sampling.
Tokens are sorted based on their assigned probabilities so that only the most likely tokens are considered. Top-k sampling directly limits the maximum number of tokens to consider, while Nucleus sampling limits the number of tokens based on the cumulative probability.
Note: The default value varies by `Model` and is specified by the`Model.top_p` attribute returned from the `getModel` function. An empty `topK` attribute indicates that the model doesn't apply top-k sampling and doesn't allow setting `topK` on requests.

`topK` `integer`

Optional. The maximum number of tokens to consider when sampling.
Gemini models use Top-p (nucleus) sampling or a combination of Top-k and nucleus sampling. Top-k sampling considers the set of `topK` most probable tokens. Models running with nucleus sampling don't allow topK setting.
Note: The default value varies by `Model` and is specified by the`Model.top_p` attribute returned from the `getModel` function. An empty `topK` attribute indicates that the model doesn't apply top-k sampling and doesn't allow setting `topK` on requests.

`seed` `integer`

Optional. Seed used in decoding. If not set, the request uses a randomly generated seed.

`presencePenalty` `number`

Optional. Presence penalty applied to the next token's logprobs if the token has already been seen in the response.
This penalty is binary on/off and not dependant on the number of times the token is used (after the first). Use `frequencyPenalty` (/api/generate-content#FIELDS.frequency_penalty) for a penalty that increases with each use.
A positive penalty will discourage the use of tokens that have already been used in the response, increasing the vocabulary.
A negative penalty will encourage the use of tokens that have already been used in the response, decreasing the vocabulary.

`frequencyPenalty` `number`

Optional. Frequency penalty applied to the next token's logprobs, multiplied by the number of times each token has been seen in the respponse so far.
A positive penalty will discourage the use of tokens that have already been used, proportional to the number of times the token has been used: The more a token is used, the more difficult it is for the model to use that token again increasing the vocabulary of responses.
Caution: A *negative* penalty will encourage the model to reuse tokens proportional to the number of times the token has been used. Small negative values will reduce the vocabulary of a response. Larger negative values will cause the model to start repeating a common token until it hits the `maxOutputTokens` (/api/generate-content#FIELDS.max_output_tokens) limit.

`responseLogprobs` `boolean`

Optional. If true, export the logprobs results in response.

`logprobs` `integer`

Optional. Only valid if `responseLogprobs=True` (/api/generate-content#FIELDS.response_logprobs). This sets the number of top logprobs to return at each decoding step in the `Candidate.logprobs_result` (/api/generate-content#FIELDS.logprobs_result). The number must be in the range of [1, 5].

`enableEnhancedCivicAnswers` `boolean`

Optional. Enables enhanced civic answers. It may not be available for all models.

`speechConfig` `object` (`SpeechConfig` (/api/generate-content#SpeechConfig))

Optional. The speech generation config.

`thinkingConfig` `object` (`ThinkingConfig` (/api/generate-content#ThinkingConfig))

Optional. Config for thinking features. An error will be returned if this field is set for models that don't support thinking.

`mediaResolution` `enum` (`MediaResolution` (/api/generate-content#MediaResolution))

Optional. If specified, the media resolution specified will be used.

JSON representation

```
{
  "stopSequences": [
    string
  ],
  "responseMimeType": string,
  "responseSchema": {
    object (Schema (/api/caching#Schema))
  },
  "responseJsonSchema": value,
  "responseModalities": [
```

JSON representation

```
    enum (Modality (/api/generate-content#Modality))
  ],
  "candidateCount": integer,
  "maxOutputTokens": integer,
  "temperature": number,
  "topP": number,
  "topK": integer,
  "seed": integer,
  "presencePenalty": number,
  "frequencyPenalty": number,
  "responseLogprobs": boolean,
  "logprobs": integer,
  "enableEnhancedCivicAnswers": boolean,
  "speechConfig": {
    object (SpeechConfig (/api/generate-content#SpeechConfig))
  },
  "thinkingConfig": {
    object (ThinkingConfig (/api/generate-content#ThinkingConfig))
  },
```

Modality

Supported modalities of the response.

Enums	
`MODALITY_UNSPECIFIED`	Default value.
`TEXT`	Indicates the model should return text.
`IMAGE`	Indicates the model should return images.
`AUDIO`	Indicates the model should return audio.

SpeechConfig

The speech generation config.

Fields

voiceConfig `object (VoiceConfig (/api/generate-content#VoiceConfig))`

The configuration in case of single-voice output.

multiSpeakerVoiceConfig

`object (MultiSpeakerVoiceConfig (/api/generate-content#MultiSpeakerVoiceConfig))`

Optional. The configuration for the multi-speaker setup. It is mutually exclusive with the voiceConfig field.

languageCode `string`

Optional. Language code (in BCP 47 format, e.g. "en-US") for speech synthesis.
Valid values are: de-DE, en-AU, en-GB, en-IN, en-US, es-US, fr-FR, hi-IN, pt-BR, ar-XA, es-ES, fr-CA, id-ID, it-IT, ja-JP, tr-TR, vi-VN, bn-IN, gu-IN, kn-IN, ml-IN, mr-IN, ta-IN, te-IN, nl-NL, ko-KR, cmn-CN, pl-PL, ru-RU, and th-TH.

JSON representation

```
{
  "voiceConfig": {
    object (VoiceConfig (/api/generate-content#VoiceConfig))
  },
  "multiSpeakerVoiceConfig": {
    object (MultiSpeakerVoiceConfig (/api/generate-content#MultiSpeakerVoiceConfig))
  },
  "languageCode": string
}
```

VoiceConfig

The configuration for the voice to use.

Fields

voice_config `Union type`

The configuration for the speaker to use. **voice_config** can be only one of the following:

> **prebuiltVoiceConfig** `object (PrebuiltVoiceConfig (/api/generate-content#PrebuiltVoiceConfig))`
>
> The configuration for the prebuilt voice to use.

JSON representation

```
{
  // voice_config
  "prebuiltVoiceConfig": {
    object (PrebuiltVoiceConfig (/api/generate-content#PrebuiltVoiceConfig))
  }
  // Union type
}
```

PrebuiltVoiceConfig

The configuration for the prebuilt speaker to use.

Fields

voiceName `string`

The name of the preset voice to use.

JSON representation

```
{
  "voiceName": string
}
```

MultiSpeakerVoiceConfig

The configuration for the multi-speaker setup.

Fields

speakerVoiceConfigs[] `object (SpeakerVoiceConfig (/api/generate-content#SpeakerVoiceConfig))`

Required. All the enabled speaker voices.

JSON representation

```
{
  "speakerVoiceConfigs": [
    {
      object (SpeakerVoiceConfig (/api/generate-content#SpeakerVoiceConfig))
    }
  ]
}
```

SpeakerVoiceConfig

The configuration for a single speaker in a multi speaker setup.

Fields

speaker `string`

Required. The name of the speaker to use. Should be the same as in the prompt.

voiceConfig `object (VoiceConfig (/api/generate-content#VoiceConfig))`

Required. The configuration for the voice to use.

JSON representation

```
{
  "speaker": string,
  "voiceConfig": {
    object (VoiceConfig (/api/generate-content#VoiceConfig))
  }
}
```

ThinkingConfig

Config for thinking features.

Fields

includeThoughts `boolean`

Indicates whether to include thoughts in the response. If true, thoughts are returned only when available.

thinkingBudget `integer`

The number of thoughts tokens that the model should generate.

JSON representation

```
{
  "includeThoughts": boolean,
  "thinkingBudget": integer
}
```

MediaResolution

Media resolution for the input media.

Enums

MEDIA_RESOLUTION_UNSPECIFIED	Media resolution has not been set.
MEDIA_RESOLUTION_LOW	Media resolution set to low (64 tokens).
MEDIA_RESOLUTION_MEDIUM	Media resolution set to medium (256 tokens).
MEDIA_RESOLUTION_HIGH	Media resolution set to high (zoomed reframing with 256 tokens).

HarmCategory

The category of a rating.

These categories cover various kinds of harms that developers may wish to adjust.

Enums

HARM_CATEGORY_UNSPECIFIED	Category is unspecified.

Enums	
HARM_CATEGORY_DEROGATORY	PaLM - Negative or harmful comments targeting identity and/or protected attribute.
HARM_CATEGORY_TOXICITY	PaLM - Content that is rude, disrespectful, or profane.
HARM_CATEGORY_VIOLENCE	PaLM - Describes scenarios depicting violence against an individual or group, or general descriptions of gore.
HARM_CATEGORY_SEXUAL	PaLM - Contains references to sexual acts or other lewd content.
HARM_CATEGORY_MEDICAL	PaLM - Promotes unchecked medical advice.
HARM_CATEGORY_DANGEROUS	PaLM - Dangerous content that promotes, facilitates, or encourages harmful acts.
HARM_CATEGORY_HARASSMENT	Gemini - Harassment content.
HARM_CATEGORY_HATE_SPEECH	Gemini - Hate speech and content.
HARM_CATEGORY_SEXUALLY_EXPLICIT	Gemini - Sexually explicit content.
HARM_CATEGORY_DANGEROUS_CONTENT	Gemini - Dangerous content.
HARM_CATEGORY_CIVIC_INTEGRITY	Gemini - Content that may be used to harm civic integrity. DEPRECATED: use enableEnhancedCivicAnswers instead.

⚠ This item is deprecated!

ModalityTokenCount

Represents token counting info for a single modality.

Fields

modality `enum` (`Modality` (/api/generate-content#Modality))

The modality associated with this token count.

`tokenCount` `integer`

Number of tokens.

JSON representation

```
{
  "modality": enum (Modality (/api/generate-content#Modality)),
  "tokenCount": integer
}
```

Modality

Content Part modality

Enums	
`MODALITY_UNSPECIFIED`	Unspecified modality.
`TEXT`	Plain text.
`IMAGE`	Image.
`VIDEO`	Video.
`AUDIO`	Audio.
`DOCUMENT`	Document, e.g. PDF.

SafetyRating

Safety rating for a piece of content.

The safety rating contains the category of harm and the harm probability level in that category for a piece of content. Content is classified for safety across a number of harm categories and the probability of the harm classification is included here.

Fields

`category` `enum` (`HarmCategory` (/api/generate-content#v1beta.HarmCategory))

Required. The category for this rating.

probability `enum` (HarmProbability (/api/generate-content#HarmProbability))

Required. The probability of harm for this content.

blocked `boolean`

Was this content blocked because of this rating?

JSON representation

```
{
  "category": enum (HarmCategory (/api/generate-content#v1beta.HarmCategory)),
  "probability": enum (HarmProbability (/api/generate-content#HarmProbability)),
  "blocked": boolean
}
```

HarmProbability

The probability that a piece of content is harmful.

The classification system gives the probability of the content being unsafe. This does not indicate the severity of harm for a piece of content.

Enums

HARM_PROBABILITY_UNSPECIFIED	Probability is unspecified.
NEGLIGIBLE	Content has a negligible chance of being unsafe.
LOW	Content has a low chance of being unsafe.
MEDIUM	Content has a medium chance of being unsafe.
HIGH	Content has a high chance of being unsafe.

SafetySetting

Safety setting, affecting the safety-blocking behavior.

Passing a safety setting for a category changes the allowed probability that content is blocked.
Fields

`category` `enum (HarmCategory (/api/generate-content#v1beta.HarmCategory))`

Required. The category for this setting.

`threshold` `enum (HarmBlockThreshold (/api/generate-content#HarmBlockThreshold))`

Required. Controls the probability threshold at which harm is blocked.

JSON representation

```
{
  "category": enum (HarmCategory (/api/generate-content#v1beta.HarmCategory)),
  "threshold": enum (HarmBlockThreshold (/api/generate-content#HarmBlockThreshold))
}
```

HarmBlockThreshold

Block at and beyond a specified harm probability.

Enums

`HARM_BLOCK_THRESHOLD_UNSPECIFIED`	Threshold is unspecified.
`BLOCK_LOW_AND_ABOVE`	Content with NEGLIGIBLE will be allowed.
`BLOCK_MEDIUM_AND_ABOVE`	Content with NEGLIGIBLE and LOW will be allowed.
`BLOCK_ONLY_HIGH`	Content with NEGLIGIBLE, LOW, and MEDIUM will be allowed.
`BLOCK_NONE`	All content will be allowed.
`OFF`	Turn off the safety filter.