**Caching**

Context caching allows you to save and reuse precomputed input tokens that you wish to use repeatedly, for example when asking different questions about the same media file. This can lead to cost and speed savings, depending on the usage. For a detailed introduction, see the [Context caching](https://ai.google.dev/gemini-api/docs/caching) guide.

**Method: cachedContents.create**

Creates CachedContent resource.

**Endpoint**

POSThttps://generativelanguage.googleapis.com/v1beta/cachedContents

**Request body**

The request body contains an instance of [CachedContent](https://ai.google.dev/api/caching" \l "CachedContent).

Fields

contents[]object ([Content](https://ai.google.dev/api/caching#Content))

Optional. Input only. Immutable. The content to cache.

tools[]object ([Tool](https://ai.google.dev/api/caching#Tool))

Optional. Input only. Immutable. A list of Tools the model may use to generate the next response

expirationUnion type

Specifies when this resource will expire. expirationcan be only one of the following:

expireTimestring ([Timestamp](https://protobuf.dev/reference/protobuf/google.protobuf/#timestamp) format)

Timestamp in UTC of when this resource is considered expired. This is *always* provided on output, regardless of what was sent on input.

Uses RFC 3339, where generated output will always be Z-normalized and uses 0, 3, 6 or 9 fractional digits. Offsets other than "Z" are also accepted. Examples: "2014-10-02T15:01:23Z", "2014-10-02T15:01:23.045123456Z" or "2014-10-02T15:01:23+05:30".

ttlstring ([Duration](https://protobuf.dev/reference/protobuf/google.protobuf/#duration) format)

Input only. New TTL for this resource, input only.

A duration in seconds with up to nine fractional digits, ending with 's'. Example: "3.5s".

displayNamestring

Optional. Immutable. The user-generated meaningful display name of the cached content. Maximum 128 Unicode characters.

modelstring

Required. Immutable. The name of the Model to use for cached content Format: models/{model}

systemInstructionobject ([Content](https://ai.google.dev/api/caching#Content))

Optional. Input only. Immutable. Developer set system instruction. Currently text only.

toolConfigobject ([ToolConfig](https://ai.google.dev/api/caching" \l "ToolConfig))

Optional. Input only. Immutable. Tool config. This config is shared for all tools.

**Example request**

[Python](https://ai.google.dev/api/caching#python)[Node.js](https://ai.google.dev/api/caching#node.js)[Go](https://ai.google.dev/api/caching#go)[Shell](https://ai.google.dev/api/caching#shell)

// Make sure to include the following import:

// import {GoogleGenAI} from '@google/genai';

const ai = new GoogleGenAI({ apiKey: process.env.GEMINI\_API\_KEY });

const filePath = path.join(media, "a11.txt");

const document = await ai.files.upload({

file: filePath,

config: { mimeType: "text/plain" },

});

console.log("Uploaded file name:", document.name);

const modelName = "gemini-1.5-flash-001";

const contents = [

createUserContent(createPartFromUri(document.uri, document.mimeType)),

];

const cache = await ai.caches.create({

model: modelName,

config: {

contents: contents,

systemInstruction: "You are an expert analyzing transcripts.",

},

});

console.log("Cache created:", cache);

const response = await ai.models.generateContent({

model: modelName,

contents: "Please summarize this transcript",

config: { cachedContent: cache.name },

});

console.log("Response text:", response.text);

**Response body**

If successful, the response body contains a newly created instance of [CachedContent](https://ai.google.dev/api/caching" \l "CachedContent).

**Method: cachedContents.list**

Lists CachedContents.

**Endpoint**

GEThttps://generativelanguage.googleapis.com/v1beta/cachedContents

**Query parameters**

pageSizeinteger

Optional. The maximum number of cached contents to return. The service may return fewer than this value. If unspecified, some default (under maximum) number of items will be returned. The maximum value is 1000; values above 1000 will be coerced to 1000.

pageTokenstring

Optional. A page token, received from a previous cachedContents.list call. Provide this to retrieve the subsequent page.

When paginating, all other parameters provided to cachedContents.list must match the call that provided the page token.

**Request body**

The request body must be empty.

**Response body**

Response with CachedContents list.

If successful, the response body contains data with the following structure:

Fields

cachedContents[]object ([CachedContent](https://ai.google.dev/api/caching" \l "CachedContent))

List of cached contents.

nextPageTokenstring

A token, which can be sent as pageToken to retrieve the next page. If this field is omitted, there are no subsequent pages.

| **JSON representation** |
| --- |
| {  "cachedContents": [  {  object ([CachedContent](https://ai.google.dev/api/caching" \l "CachedContent))  }  ],  "nextPageToken": string  } |

**Method: cachedContents.get**

Reads CachedContent resource.

**Endpoint**

GEThttps://generativelanguage.googleapis.com/v1beta/{name=cachedContents/\*}

**Path parameters**

namestring

Required. The resource name referring to the content cache entry. Format: cachedContents/{id} It takes the form cachedContents/{cachedcontent}.

**Request body**

The request body must be empty.

**Example request**

[Python](https://ai.google.dev/api/caching#python)[Node.js](https://ai.google.dev/api/caching#node.js)[Go](https://ai.google.dev/api/caching#go)[Shell](https://ai.google.dev/api/caching#shell)

// Make sure to include the following import:

// import {GoogleGenAI} from '@google/genai';

const ai = new GoogleGenAI({ apiKey: process.env.GEMINI\_API\_KEY });

const filePath = path.join(media, "a11.txt");

const document = await ai.files.upload({

file: filePath,

config: { mimeType: "text/plain" },

});

console.log("Uploaded file name:", document.name);

const modelName = "gemini-1.5-flash-001";

const contents = [

createUserContent(createPartFromUri(document.uri, document.mimeType)),

];

const cache = await ai.caches.create({

model: modelName,

config: {

contents: contents,

systemInstruction: "You are an expert analyzing transcripts.",

},

});

const retrievedCache = await ai.caches.get({ name: cache.name });

console.log("Retrieved Cache:", retrievedCache);

**Response body**

If successful, the response body contains an instance of [CachedContent](https://ai.google.dev/api/caching" \l "CachedContent).

**Method: cachedContents.patch**

Updates CachedContent resource (only expiration is updatable).

**Endpoint**

PATCHhttps://generativelanguage.googleapis.com/v1beta/{cachedContent.name=cachedContents/\*}

PATCH https://generativelanguage.googleapis.com/v1beta/{cachedContent.name=cachedContents/\*}

**Path parameters**

cachedContent.namestring

Output only. Identifier. The resource name referring to the cached content. Format: cachedContents/{id} It takes the form cachedContents/{cachedcontent}.

**Query parameters**

updateMaskstring ([FieldMask](https://protobuf.dev/reference/protobuf/google.protobuf/" \l "field-mask) format)

The list of fields to update.

This is a comma-separated list of fully qualified names of fields. Example: "user.displayName,photo".

**Request body**

The request body contains an instance of [CachedContent](https://ai.google.dev/api/caching" \l "CachedContent).

Fields

expirationUnion type

Specifies when this resource will expire. expirationcan be only one of the following:

expireTimestring ([Timestamp](https://protobuf.dev/reference/protobuf/google.protobuf/#timestamp) format)

Timestamp in UTC of when this resource is considered expired. This is *always* provided on output, regardless of what was sent on input.

Uses RFC 3339, where generated output will always be Z-normalized and uses 0, 3, 6 or 9 fractional digits. Offsets other than "Z" are also accepted. Examples: "2014-10-02T15:01:23Z", "2014-10-02T15:01:23.045123456Z" or "2014-10-02T15:01:23+05:30".

ttlstring ([Duration](https://protobuf.dev/reference/protobuf/google.protobuf/#duration) format)

Input only. New TTL for this resource, input only.

A duration in seconds with up to nine fractional digits, ending with 's'. Example: "3.5s".

**Example request**

[Python](https://ai.google.dev/api/caching#python)[Node.js](https://ai.google.dev/api/caching#node.js)[Go](https://ai.google.dev/api/caching#go)[Shell](https://ai.google.dev/api/caching#shell)

// Make sure to include the following import:

// import {GoogleGenAI} from '@google/genai';

const ai = new GoogleGenAI({ apiKey: process.env.GEMINI\_API\_KEY });

const filePath = path.join(media, "a11.txt");

const document = await ai.files.upload({

file: filePath,

config: { mimeType: "text/plain" },

});

console.log("Uploaded file name:", document.name);

const modelName = "gemini-1.5-flash-001";

const contents = [

createUserContent(createPartFromUri(document.uri, document.mimeType)),

];

let cache = await ai.caches.create({

model: modelName,

config: {

contents: contents,

systemInstruction: "You are an expert analyzing transcripts.",

},

});

// Update the cache's time-to-live (ttl)

const ttl = `${2 \* 3600}s`; // 2 hours in seconds

cache = await ai.caches.update({

name: cache.name,

config: { ttl },

});

console.log("After update (TTL):", cache);

// Alternatively, update the expire\_time directly (in RFC 3339 format with a "Z" suffix)

const expireTime = new Date(Date.now() + 15 \* 60000)

.toISOString()

.replace(/\.\d{3}Z$/, "Z");

cache = await ai.caches.update({

name: cache.name,

config: { expireTime: expireTime },

});

console.log("After update (expire\_time):", cache);

**Response body**

If successful, the response body contains an instance of [CachedContent](https://ai.google.dev/api/caching" \l "CachedContent).

**Method: cachedContents.delete**

Deletes CachedContent resource.

**Endpoint**

DELETEhttps://generativelanguage.googleapis.com/v1beta/{name=cachedContents/\*}

**Path parameters**

namestring

Required. The resource name referring to the content cache entry Format: cachedContents/{id} It takes the form cachedContents/{cachedcontent}.

**Request body**

The request body must be empty.

**Example request**

[Python](https://ai.google.dev/api/caching#python)[Node.js](https://ai.google.dev/api/caching#node.js)[Go](https://ai.google.dev/api/caching#go)[Shell](https://ai.google.dev/api/caching#shell)

// Make sure to include the following import:

// import {GoogleGenAI} from '@google/genai';

const ai = new GoogleGenAI({ apiKey: process.env.GEMINI\_API\_KEY });

const filePath = path.join(media, "a11.txt");

const document = await ai.files.upload({

file: filePath,

config: { mimeType: "text/plain" },

});

console.log("Uploaded file name:", document.name);

const modelName = "gemini-1.5-flash-001";

const contents = [

createUserContent(createPartFromUri(document.uri, document.mimeType)),

];

const cache = await ai.caches.create({

model: modelName,

config: {

contents: contents,

systemInstruction: "You are an expert analyzing transcripts.",

},

});

await ai.caches.delete({ name: cache.name });

console.log("Cache deleted:", cache.name);

**Response body**

If successful, the response body is an empty JSON object.

**REST Resource: cachedContents**

**Resource: CachedContent**

Content that has been preprocessed and can be used in subsequent request to GenerativeService.

Cached content can be only used with model it was created for.

Fields

contents[]object ([Content](https://ai.google.dev/api/caching#Content))

Optional. Input only. Immutable. The content to cache.

tools[]object ([Tool](https://ai.google.dev/api/caching#Tool))

Optional. Input only. Immutable. A list of Tools the model may use to generate the next response

createTimestring ([Timestamp](https://protobuf.dev/reference/protobuf/google.protobuf/#timestamp) format)

Output only. Creation time of the cache entry.

Uses RFC 3339, where generated output will always be Z-normalized and uses 0, 3, 6 or 9 fractional digits. Offsets other than "Z" are also accepted. Examples: "2014-10-02T15:01:23Z", "2014-10-02T15:01:23.045123456Z" or "2014-10-02T15:01:23+05:30".

updateTimestring ([Timestamp](https://protobuf.dev/reference/protobuf/google.protobuf/#timestamp) format)

Output only. When the cache entry was last updated in UTC time.

Uses RFC 3339, where generated output will always be Z-normalized and uses 0, 3, 6 or 9 fractional digits. Offsets other than "Z" are also accepted. Examples: "2014-10-02T15:01:23Z", "2014-10-02T15:01:23.045123456Z" or "2014-10-02T15:01:23+05:30".

usageMetadataobject ([UsageMetadata](https://ai.google.dev/api/caching" \l "UsageMetadata))

Output only. Metadata on the usage of the cached content.

expirationUnion type

Specifies when this resource will expire. expirationcan be only one of the following:

expireTimestring ([Timestamp](https://protobuf.dev/reference/protobuf/google.protobuf/#timestamp) format)

Timestamp in UTC of when this resource is considered expired. This is *always* provided on output, regardless of what was sent on input.

Uses RFC 3339, where generated output will always be Z-normalized and uses 0, 3, 6 or 9 fractional digits. Offsets other than "Z" are also accepted. Examples: "2014-10-02T15:01:23Z", "2014-10-02T15:01:23.045123456Z" or "2014-10-02T15:01:23+05:30".

ttlstring ([Duration](https://protobuf.dev/reference/protobuf/google.protobuf/#duration) format)

Input only. New TTL for this resource, input only.

A duration in seconds with up to nine fractional digits, ending with 's'. Example: "3.5s".

namestring

Output only. Identifier. The resource name referring to the cached content. Format: cachedContents/{id}

displayNamestring

Optional. Immutable. The user-generated meaningful display name of the cached content. Maximum 128 Unicode characters.

modelstring

Required. Immutable. The name of the Model to use for cached content Format: models/{model}

systemInstructionobject ([Content](https://ai.google.dev/api/caching#Content))

Optional. Input only. Immutable. Developer set system instruction. Currently text only.

toolConfigobject ([ToolConfig](https://ai.google.dev/api/caching" \l "ToolConfig))

Optional. Input only. Immutable. Tool config. This config is shared for all tools.

| **JSON representation** |
| --- |
| {  "contents": [  {  object ([Content](https://ai.google.dev/api/caching#Content))  }  ],  "tools": [  {  object ([Tool](https://ai.google.dev/api/caching#Tool))  }  ],  "createTime": string,  "updateTime": string,  "usageMetadata": {  object ([UsageMetadata](https://ai.google.dev/api/caching" \l "UsageMetadata))  },  // expiration  "expireTime": string,  "ttl": string  // Union type  "name": string,  "displayName": string,  "model": string,  "systemInstruction": {  object ([Content](https://ai.google.dev/api/caching#Content))  },  "toolConfig": {  object ([ToolConfig](https://ai.google.dev/api/caching" \l "ToolConfig))  }  } |

**Content**

The base structured datatype containing multi-part content of a message.

A Content includes a role field designating the producer of the Content and a parts field containing multi-part data that contains the content of the message turn.

Fields

parts[]object ([Part](https://ai.google.dev/api/caching#Part))

Ordered Parts that constitute a single message. Parts may have different MIME types.

rolestring

Optional. The producer of the content. Must be either 'user' or 'model'.

Useful to set for multi-turn conversations, otherwise can be left blank or unset.

| **JSON representation** |
| --- |
| {  "parts": [  {  object ([Part](https://ai.google.dev/api/caching#Part))  }  ],  "role": string  } |

**Part**

A datatype containing media that is part of a multi-part Content message.

A Part consists of data which has an associated datatype. A Part can only contain one of the accepted types in Part.data.

A Part must have a fixed IANA MIME type identifying the type and subtype of the media if the inlineData field is filled with raw bytes.

Fields

thoughtboolean

Optional. Indicates if the part is thought from the model.

dataUnion type

data can be only one of the following:

textstring

Inline text.

inlineDataobject ([Blob](https://ai.google.dev/api/caching#Blob))

Inline media bytes.

functionCallobject ([FunctionCall](https://ai.google.dev/api/caching" \l "FunctionCall))

A predicted FunctionCall returned from the model that contains a string representing the FunctionDeclaration.name with the arguments and their values.

functionResponseobject ([FunctionResponse](https://ai.google.dev/api/caching" \l "FunctionResponse))

The result output of a FunctionCall that contains a string representing the FunctionDeclaration.name and a structured JSON object containing any output from the function is used as context to the model.

fileDataobject ([FileData](https://ai.google.dev/api/caching#FileData))

URI based data.

executableCodeobject ([ExecutableCode](https://ai.google.dev/api/caching" \l "ExecutableCode))

Code generated by the model that is meant to be executed.

codeExecutionResultobject ([CodeExecutionResult](https://ai.google.dev/api/caching" \l "CodeExecutionResult))

Result of executing the ExecutableCode.

| **JSON representation** |
| --- |
| {  "thought": boolean,  // data  "text": string,  "inlineData": {  object ([Blob](https://ai.google.dev/api/caching#Blob))  },  "functionCall": {  object ([FunctionCall](https://ai.google.dev/api/caching" \l "FunctionCall))  },  "functionResponse": {  object ([FunctionResponse](https://ai.google.dev/api/caching" \l "FunctionResponse))  },  "fileData": {  object ([FileData](https://ai.google.dev/api/caching#FileData))  },  "executableCode": {  object ([ExecutableCode](https://ai.google.dev/api/caching" \l "ExecutableCode))  },  "codeExecutionResult": {  object ([CodeExecutionResult](https://ai.google.dev/api/caching" \l "CodeExecutionResult))  }  // Union type  } |

**Blob**

Raw media bytes.

Text should not be sent as raw bytes, use the 'text' field.

Fields

mimeTypestring

The IANA standard MIME type of the source data. Examples: - image/png - image/jpeg If an unsupported MIME type is provided, an error will be returned. For a complete list of supported types, see [Supported file formats](https://ai.google.dev/gemini-api/docs/prompting_with_media#supported_file_formats).

datastring ([bytes](https://developers.google.com/discovery/v1/type-format) format)

Raw bytes for media formats.

A base64-encoded string.

| **JSON representation** |
| --- |
| {  "mimeType": string,  "data": string  } |

**FunctionCall**

A predicted FunctionCall returned from the model that contains a string representing the FunctionDeclaration.name with the arguments and their values.

Fields

idstring

Optional. The unique id of the function call. If populated, the client to execute the functionCall and return the response with the matching id.

namestring

Required. The name of the function to call. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 63.

argsobject ([Struct](https://protobuf.dev/reference/protobuf/google.protobuf/#struct) format)

Optional. The function parameters and values in JSON object format.

| **JSON representation** |
| --- |
| {  "id": string,  "name": string,  "args": {  object  }  } |

**FunctionResponse**

The result output from a FunctionCall that contains a string representing the FunctionDeclaration.name and a structured JSON object containing any output from the function is used as context to the model. This should contain the result of aFunctionCall made based on model prediction.

Fields

idstring

Optional. The id of the function call this response is for. Populated by the client to match the corresponding function call id.

namestring

Required. The name of the function to call. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 63.

responseobject ([Struct](https://protobuf.dev/reference/protobuf/google.protobuf/#struct) format)

Required. The function response in JSON object format.

| **JSON representation** |
| --- |
| {  "id": string,  "name": string,  "response": {  object  }  } |

**FileData**

URI based data.

Fields

mimeTypestring

Optional. The IANA standard MIME type of the source data.

fileUristring

Required. URI.

| **JSON representation** |
| --- |
| {  "mimeType": string,  "fileUri": string  } |

**ExecutableCode**

Code generated by the model that is meant to be executed, and the result returned to the model.

Only generated when using the CodeExecution tool, in which the code will be automatically executed, and a corresponding CodeExecutionResult will also be generated.

Fields

languageenum ([Language](https://ai.google.dev/api/caching#Language))

Required. Programming language of the code.

codestring

Required. The code to be executed.

| **JSON representation** |
| --- |
| {  "language": enum ([Language](https://ai.google.dev/api/caching#Language)),  "code": string  } |

**Language**

Supported programming languages for the generated code.

| **Enums** | |
| --- | --- |
| LANGUAGE\_UNSPECIFIED | Unspecified language. This value should not be used. |
| PYTHON | Python >= 3.10, with numpy and simpy available. |

**CodeExecutionResult**

Result of executing the ExecutableCode.

Only generated when using the CodeExecution, and always follows a part containing the ExecutableCode.

Fields

outcomeenum ([Outcome](https://ai.google.dev/api/caching#Outcome))

Required. Outcome of the code execution.

outputstring

Optional. Contains stdout when code execution is successful, stderr or other description otherwise.

| **JSON representation** |
| --- |
| {  "outcome": enum ([Outcome](https://ai.google.dev/api/caching#Outcome)),  "output": string  } |

**Outcome**

Enumeration of possible outcomes of the code execution.

| **Enums** | |
| --- | --- |
| OUTCOME\_UNSPECIFIED | Unspecified status. This value should not be used. |
| OUTCOME\_OK | Code execution completed successfully. |
| OUTCOME\_FAILED | Code execution finished but with a failure. stderr should contain the reason. |
| OUTCOME\_DEADLINE\_EXCEEDED | Code execution ran for too long, and was cancelled. There may or may not be a partial output present. |

**Tool**

Tool details that the model may use to generate 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.

Fields

functionDeclarations[]object ([FunctionDeclaration](https://ai.google.dev/api/caching" \l "FunctionDeclaration))

Optional. A list of FunctionDeclarations available to the model that can be used for function calling.

The model or system does not execute the function. Instead the defined function may be returned as a [FunctionCall](https://ai.google.dev/api/caching" \l "Part.FIELDS.function_call) with arguments to the client side for execution. The model may decide to call a subset of these functions by populating [FunctionCall](https://ai.google.dev/api/caching" \l "Part.FIELDS.function_call) in the response. The next conversation turn may contain a [FunctionResponse](https://ai.google.dev/api/caching" \l "Part.FIELDS.function_response) with the [Content.role](https://ai.google.dev/api/caching" \l "Content.FIELDS.role) "function" generation context for the next model turn.

googleSearchRetrievalobject ([GoogleSearchRetrieval](https://ai.google.dev/api/caching" \l "GoogleSearchRetrieval))

Optional. Retrieval tool that is powered by Google search.

codeExecutionobject ([CodeExecution](https://ai.google.dev/api/caching" \l "CodeExecution))

Optional. Enables the model to execute code as part of generation.

googleSearchobject ([GoogleSearch](https://ai.google.dev/api/caching" \l "GoogleSearch))

Optional. GoogleSearch tool type. Tool to support Google Search in Model. Powered by Google.

| **JSON representation** |
| --- |
| {  "functionDeclarations": [  {  object ([FunctionDeclaration](https://ai.google.dev/api/caching" \l "FunctionDeclaration))  }  ],  "googleSearchRetrieval": {  object ([GoogleSearchRetrieval](https://ai.google.dev/api/caching" \l "GoogleSearchRetrieval))  },  "codeExecution": {  object ([CodeExecution](https://ai.google.dev/api/caching" \l "CodeExecution))  },  "googleSearch": {  object ([GoogleSearch](https://ai.google.dev/api/caching" \l "GoogleSearch))  }  } |

**FunctionDeclaration**

Structured representation of a function declaration as defined by the [OpenAPI 3.03 specification](https://spec.openapis.org/oas/v3.0.3). Included in this declaration are the function name and parameters. This FunctionDeclaration is a representation of a block of code that can be used as a Tool by the model and executed by the client.

Fields

namestring

Required. The name of the function. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 63.

descriptionstring

Required. A brief description of the function.

parametersobject ([Schema](https://ai.google.dev/api/caching#Schema))

Optional. Describes the parameters to this function. Reflects the Open API 3.03 Parameter Object string Key: the name of the parameter. Parameter names are case sensitive. Schema Value: the Schema defining the type used for the parameter.

responseobject ([Schema](https://ai.google.dev/api/caching#Schema))

Optional. Describes the output from this function in JSON Schema format. Reflects the Open API 3.03 Response Object. The Schema defines the type used for the response value of the function.

| **JSON representation** |
| --- |
| {  "name": string,  "description": string,  "parameters": {  object ([Schema](https://ai.google.dev/api/caching#Schema))  },  "response": {  object ([Schema](https://ai.google.dev/api/caching#Schema))  }  } |

**Schema**

The Schema object allows the definition of input and output data types. These types can be objects, but also primitives and arrays. Represents a select subset of an [OpenAPI 3.0 schema object](https://spec.openapis.org/oas/v3.0.3" \l "schema).

Fields

typeenum ([Type](https://ai.google.dev/api/caching#Type))

Required. Data type.

formatstring

Optional. The format of the data. This is used only for primitive datatypes. Supported formats: for NUMBER type: float, double for INTEGER type: int32, int64 for STRING type: enum, date-time

titlestring

Optional. The title of the schema.

descriptionstring

Optional. A brief description of the parameter. This could contain examples of use. Parameter description may be formatted as Markdown.

nullableboolean

Optional. Indicates if the value may be null.

enum[]string

Optional. Possible values of the element of Type.STRING with enum format. For example we can define an Enum Direction as : {type:STRING, format:enum, enum:["EAST", NORTH", "SOUTH", "WEST"]}

maxItemsstring ([int64](https://developers.google.com/discovery/v1/type-format) format)

Optional. Maximum number of the elements for Type.ARRAY.

minItemsstring ([int64](https://developers.google.com/discovery/v1/type-format) format)

Optional. Minimum number of the elements for Type.ARRAY.

propertiesmap (key: string, value: object ([Schema](https://ai.google.dev/api/caching#Schema)))

Optional. Properties of Type.OBJECT.

An object containing a list of "key": value pairs. Example: { "name": "wrench", "mass": "1.3kg", "count": "3" }.

required[]string

Optional. Required properties of Type.OBJECT.

minPropertiesstring ([int64](https://developers.google.com/discovery/v1/type-format) format)

Optional. Minimum number of the properties for Type.OBJECT.

maxPropertiesstring ([int64](https://developers.google.com/discovery/v1/type-format) format)

Optional. Maximum number of the properties for Type.OBJECT.

minLengthstring ([int64](https://developers.google.com/discovery/v1/type-format) format)

Optional. SCHEMA FIELDS FOR TYPE STRING Minimum length of the Type.STRING

maxLengthstring ([int64](https://developers.google.com/discovery/v1/type-format) format)

Optional. Maximum length of the Type.STRING

patternstring

Optional. Pattern of the Type.STRING to restrict a string to a regular expression.

examplevalue ([Value](https://protobuf.dev/reference/protobuf/google.protobuf/#value) format)

Optional. Example of the object. Will only populated when the object is the root.

anyOf[]object ([Schema](https://ai.google.dev/api/caching#Schema))

Optional. The value should be validated against any (one or more) of the subschemas in the list.

propertyOrdering[]string

Optional. The order of the properties. Not a standard field in open api spec. Used to determine the order of the properties in the response.

defaultvalue ([Value](https://protobuf.dev/reference/protobuf/google.protobuf/#value) format)

Optional. Default value of the field. Per JSON Schema, this field is intended for documentation generators and doesn't affect validation. Thus it's included here and ignored so that developers who send schemas with a default field don't get unknown-field errors.

itemsobject ([Schema](https://ai.google.dev/api/caching#Schema))

Optional. Schema of the elements of Type.ARRAY.

minimumnumber

Optional. SCHEMA FIELDS FOR TYPE INTEGER and NUMBER Minimum value of the Type.INTEGER and Type.NUMBER

maximumnumber

Optional. Maximum value of the Type.INTEGER and Type.NUMBER

| **JSON representation** |
| --- |
| {  "type": enum ([Type](https://ai.google.dev/api/caching#Type)),  "format": string,  "title": string,  "description": string,  "nullable": boolean,  "enum": [  string  ],  "maxItems": string,  "minItems": string,  "properties": {  string: {  object ([Schema](https://ai.google.dev/api/caching#Schema))  },  ...  },  "required": [  string  ],  "minProperties": string,  "maxProperties": string,  "minLength": string,  "maxLength": string,  "pattern": string,  "example": value,  "anyOf": [  {  object ([Schema](https://ai.google.dev/api/caching#Schema))  }  ],  "propertyOrdering": [  string  ],  "default": value,  "items": {  object ([Schema](https://ai.google.dev/api/caching#Schema))  },  "minimum": number,  "maximum": number  } |

**Type**

Type contains the list of OpenAPI data types as defined by <https://spec.openapis.org/oas/v3.0.3#data-types>

| **Enums** | |
| --- | --- |
| TYPE\_UNSPECIFIED | Not specified, should not be used. |
| STRING | String type. |
| NUMBER | Number type. |
| INTEGER | Integer type. |
| BOOLEAN | Boolean type. |
| ARRAY | Array type. |
| OBJECT | Object type. |
| NULL | Null type. |

**GoogleSearchRetrieval**

Tool to retrieve public web data for grounding, powered by Google.

Fields

dynamicRetrievalConfigobject ([DynamicRetrievalConfig](https://ai.google.dev/api/caching" \l "DynamicRetrievalConfig))

Specifies the dynamic retrieval configuration for the given source.

| **JSON representation** |
| --- |
| {  "dynamicRetrievalConfig": {  object ([DynamicRetrievalConfig](https://ai.google.dev/api/caching" \l "DynamicRetrievalConfig))  }  } |

**DynamicRetrievalConfig**

Describes the options to customize dynamic retrieval.

Fields

modeenum ([Mode](https://ai.google.dev/api/caching#Mode))

The mode of the predictor to be used in dynamic retrieval.

dynamicThresholdnumber

The threshold to be used in dynamic retrieval. If not set, a system default value is used.

| **JSON representation** |
| --- |
| {  "mode": enum ([Mode](https://ai.google.dev/api/caching#Mode)),  "dynamicThreshold": number  } |

**Mode**

The mode of the predictor to be used in dynamic retrieval.

| **Enums** | |
| --- | --- |
| MODE\_UNSPECIFIED | Always trigger retrieval. |
| MODE\_DYNAMIC | Run retrieval only when system decides it is necessary. |

**CodeExecution**

This type has no fields.

Tool that executes code generated by the model, and automatically returns the result to the model.

See also ExecutableCode and CodeExecutionResult which are only generated when using this tool.

**GoogleSearch**

This type has no fields.

GoogleSearch tool type. Tool to support Google Search in Model. Powered by Google.

**ToolConfig**

The Tool configuration containing parameters for specifying Tool use in the request.

Fields

functionCallingConfigobject ([FunctionCallingConfig](https://ai.google.dev/api/caching" \l "FunctionCallingConfig))

Optional. Function calling config.

| **JSON representation** |
| --- |
| {  "functionCallingConfig": {  object ([FunctionCallingConfig](https://ai.google.dev/api/caching" \l "FunctionCallingConfig))  }  } |

**FunctionCallingConfig**

Configuration for specifying function calling behavior.

Fields

modeenum ([Mode](https://ai.google.dev/api/caching#Mode_1))

Optional. Specifies the mode in which function calling should execute. If unspecified, the default value will be set to AUTO.

allowedFunctionNames[]string

Optional. A set of function names that, when provided, limits the functions the model will call.

This should only be set when the Mode is ANY. Function names should match [FunctionDeclaration.name]. With mode set to ANY, model will predict a function call from the set of function names provided.

| **JSON representation** |
| --- |
| {  "mode": enum ([Mode](https://ai.google.dev/api/caching#Mode_1)),  "allowedFunctionNames": [  string  ]  } |

**Mode**

Defines the execution behavior for function calling by defining the execution mode.

| **Enums** | |
| --- | --- |
| MODE\_UNSPECIFIED | Unspecified function calling mode. This value should not be used. |
| AUTO | Default model behavior, model decides to predict either a function call or a natural language response. |
| ANY | Model is constrained to always predicting a function call only. If "allowedFunctionNames" are set, the predicted function call will be limited to any one of "allowedFunctionNames", else the predicted function call will be any one of the provided "functionDeclarations". |
| NONE | Model will not predict any function call. Model behavior is same as when not passing any function declarations. |
| VALIDATED | Model decides to predict either a function call or a natural language response, but will validate function calls with constrained decoding. |

**UsageMetadata**

Metadata on the usage of the cached content.

Fields

totalTokenCountinteger

Total number of tokens that the cached content consumes.

| **JSON representation** |
| --- |
| {  "totalTokenCount": integer  } |

Was